Final Report RSC 3556
NASA CR
160350

THE SYSTEM AND HARDWARE DESIGN OF REAL-TIME FAN BEAM SCATTEROMETER DATA PROCESSORS

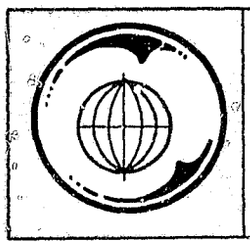
By

John P. Claassen
Robert O. Stroud
Billy V. Clark
B. Randall Jean
Richard W. Newton

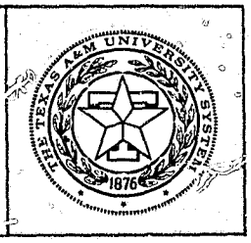
March 1979

Principal Investigator: Richard W. Newton
Program Manager: John P. Claassen

Sponsored by:
National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Contract NAS9-15311



TEXAS A&M UNIVERSITY
REMOTE SENSING CENTER
COLLEGE STATION, TEXAS



Final Report RSC 3556

THE SYSTEM AND HARDWARE DESIGN OF REAL-TIME FAN BEAM SCATTEROMETER DATA PROCESSORS

By

John P. Claassen
Robert O. Stroud
Billy V. Clark
B. Randall Jean
Richard W. Newton

March 1979

Sponsored by:

National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Contract NAS9-15311

1

TABLE OF CONTENTS

II

LIST OF FIGURES

LIST OF TABLES

THE SYSTEM AND HARDWARE DESIGN OF REAL-TIME FAN BEAM SCATTEROMETER DATA PROCESSORS

1.0 SUMMARY

The Remote Sensing Center at Texas A&M University (TAMU) has completed a major system design effort for the National Aeronautics and Space Administration, Johnson Space Center (NASA/JSC) for a class of real-time radar scatterometer signal processors. A system design is presented which employs state-of-the-art integrated circuits and signal processing technology. This report presents the details of the design of a system specifically tailored for the C- and L-band radar scatterometer systems. The design is sufficiently flexible, however, to accommodate any scatterometer system with only minor modifications in sampling frequencies and filter bandwidths.

The design is based upon a 512 stage CCD (change coupled device) analog transversal filter which is employed to perform a 512 point discrete Fourier transform (DFT) via the chirp Z-transform (CZT) algorithm. The CZT implementation provides for rapid power spectral density computations on the radar return signal from which the normalized radar cross-section α can be computed for a number of desired incident angles at a real-time rate.

The scatterometer processing system design is presented in sufficient detail to allow for construction of an engineering test mode of the processor directly from the design data contained in this report. Breadboard testing of key elements of the system have proven the validity and applicability of the design.

2.0 INTRODUCTION

2.1 Historical Background

The NASA Johnson Space Center has acquired a series of airborne fan beam scatterometers since the mid 1960's for support of various earth and space related programs. Data from an early system proved helpful in identifying an ocean wind measurement technique. This work eventually led to the scatterometer system aboard SEASAT A. Raw scatterometer data from these early experiments were post processed into quantitative engineering units using a general purpose computer. The time and expense required to process data caused the delay between data product delivery and data acquisition to be excessive. As a result, utilization of the scatterometers were eventually removed from service.

More recently, radar data requests by the NASA soil moisture program provided enough justification to warrant reinstating the 0.4 GHz, 1.6 GHz, and 13.3 GHz scatterometers into service, and constructing a new 4.75 GHz scatterometer. The earlier experiences with scatterometer data processing led to a program of developing faster, cheaper methods of data handling and processing. The initial thrust of the program produced a demonstration processor for use with the 13.3 GHz scatterometer. The philosophy behind this processor was to provide a real-time quick look capability for verifying data characteristics (i.e., system operation) and 2) provide a method of identifying those data to be post processed on a large computer to the accuracies needed for analysis. This processor was developed and constructed at Texas A&M University (TAMU) under NASA contract [1] to process a single polarization channel over a limited set of viewing angles. In

addition to this hardware processor, TAMU developed software routines for utilization on general purpose computers to reduce the raw data needed for analysis to calibrated engineering units.

Recent technological advances have suggested that by combining analog and digital processing methods into a single processor, real-time processing and real-time rate post time processing of scatterometer data to calibrated engineering units could be accomplished. Such a system could provide all of the capability in terms of viewing angles, resolution and adaptability that the post time software systems previously developed could provide, with a potential for more accurate results as a result of eliminating the analog recorder when operating in a real-time mode. Such a system would provide experimenters with calibrated data on a timely basis with many fewer manhours from data flight to delivery. This realization provided the basis for the current efforts reported in this document.

2.2 Design Objective and Overview

Designs for two airborne radar scatterometer processors for use with the NASA 1.6 GHz and 4.75 GHz scatterometers were identified and analyzed. A portion of the processor was implemented to evaluate a "state-of-the-art" component proposed for use in the processor. This component permitted a standardized design approach which is extendable to other NASA fan beam scatterometers. The current effort exploited design experiences from previous hardware and software processors to minimize significant error contributions and to assure repeatability in performance. However, innovations were also introduced as a result of the hybrid sampled analog and digital approach to provide a flexible operator/experimenter oriented

system. As a result of these new insights, major improvements were also identified for use in the purely software approaches to processing scatterometer data. This benefitted another NASA sponsored program to develop a more efficient software routine to process scatterometer data on an interim basis while the hardware processors undergo development. This latter effort ran concurrently with this program and afforded an opportunity to also test, anticipate and prove the characteristics of the hardware design described within this report.

The hardware design features a chirp Z-transform (CZT) approach to filtering the Doppler spread radar return. The CZT is implemented with multiplying digital to analog converters and a charge coupled transversal filter. The filtering operation reduces to that of performing a discrete Fourier transform (DFT) of the radar return when represented in complex valued form. As a consequence, no Hilbert transform operation (sign sensing) is required to separate fore and aft returns. Both are provided simultaneously with considerable reduction in complexity. The subsequent processing, is actually limited to the aft data; however, the fore data is available within the processor should future efforts require it.

There are many advantages in the CZT approach. It permits high frequency resolution of the Doppler return. As a consequence, the return may be measured with good angular resolution. This also permits the processor to adapt with changes in aircraft velocity to tract the desired viewing angles by simply using a different set of spectral outputs. It will also permit arbitrary choices in viewing angles if desired. The CZT approach can be readily applied to scatterometers operating at other wavelengths primarily by altering the sampling frequency.

The power spectral density (PSD) of the total return is formed from the chirp Z transformed data. The formation of the PSD requires that the spectral data be detected (squared) and accumulated (averaged) over a period of time. To achieve the accuracy and the dynamic range required in scatterometry the detection and accumulation is accomplished digitally.

The detected and averaged data is converted to estimates of the scattering coefficients σ^0 at eight viewing angles over the aft sector. The conversion is implemented in software and requires the application of radar range, pattern data, viewing angle, and transmitted power to yield a calibrated result. In addition, the software permits interactive control of the processor. Since the computations and control are provided by software, any portion of the operating system can be altered should the need arise.

The design approach was partially evaluated by actually implementing a subsystem of the scatterometer processor. An evaluation to this detail was required to validate the performance and dynamic range of the charge coupled devices and associated circuitry since this is a "state-of-the-art" item.

This report describes the system design theory, the system operating rationale and architecture, the hardware and software designs and an evaluation of the CZT approach for the scatterometer processors profiled above. In particular, Section 3.0 develops the system design theory background. The characteristics of CW fan beam scatterometers are related to the scatterometer equation to identify the measurement theory. It is shown that the angular scattering characteristic can be resolved by estimating the PSD of the radar return. The precision by which the PSD

estimated is related to the bandwidth - time product by analogy with classical fading theory. The technique by which the fore and aft spectra are separated using a DFT method is then identified. The DFT is related to the CZT and the means by which the CZT may be implemented is then established.

Section 4.0 is dedicated to establishing a suitable operating rationale for the processor. Trade-offs between angular resolution, ground resolution, precision, and beam resolution are established and evaluated to identify a suitable operating mode to satisfy user requirements and system constraints.

Section 5.0 describes in detail the system architecture and carefully distinguishes between the target system and the engineering development model proposed within this effort. An overview of the internal operation of the system is also presented.

Section 6.0 discloses the hardware design rationale for the PSD estimation subsystem. The results of evaluating a breadboard model of the PSD subsystem is also included.

The micro-processing subsystem and its interface units are treated in Section 7.0. A high speed floating point processor manufactured by Advanced Micro-Computer Corporation (AMC) is identified as a potential replacement for the 80/20 identified in work prior to this contract. The AMC mono-board computer is fully compatible with the 80/20 system but is much faster and permits efficient programming. The design and operation of three interface units which allow communication with the micro-processor are also described.

Section 8.0 is devoted to the system controller and processing software. There the software design rationale, architecture and operating procedure are described.

The conclusions and recommendations appear in Section 9.0. The entire document is supported by a number of appendices. In addition to treating special topics, the appendices serve to preserve parts listings for each subassembly and vendors literature on special components. Schematics are provided in blue print form in a special container. Special 8080 software routines required to run and evaluate the CZT demonstration processor are also included in the appendices.

3.0 DESIGN THEORY

3.1 Introduction

Airborne fan beam scatterometers permit simultaneous backscatter observations over a range of incident angles. By confining the antenna beam width in the crosstrack dimension and spreading the beam in the along track dimension, Doppler filtering may be employed in a CW system to resolve the average return power at various incident angles, each of which is spanned by a small angular window as illustrated by Figure 3.1. Combinations of transmit and receive polarizations permit like and cross polarized scattering properties of distributed targets to be measured. When the aircraft is flown over the same distributed target at different headings about the compass, the azimuthal as well as the incident angular behaviors may be documented.

3.2 The Scatterometer Equation and Fan Beam Systems

For a large class of distributed targets the returns from slightly different angular directions are essentially uncorrelated. Where a particular direction is denoted by (θ,ϕ) within the coordinate system of Figure 3.2, the total return power may be described by summing returns from patches of the target located in various angular directions (θ_i,ϕ_j). If the radar cross section in direction (θ_i,ϕ_j) is denoted by $\sigma_{pq}(\theta_i,\phi_j)$, the total return power may be expressed as

$$W_r' = \frac{\lambda^2}{(4\pi)^3} W_t \sum_{i=1}^{N} \sum_{j=1}^{M} G_{tp}(\theta_i,\phi_j) G_{rq}(\theta_i,\phi_j) \sigma_{pq}(\theta_i,\phi_j)/R_i^4 \qquad (3.1)$$

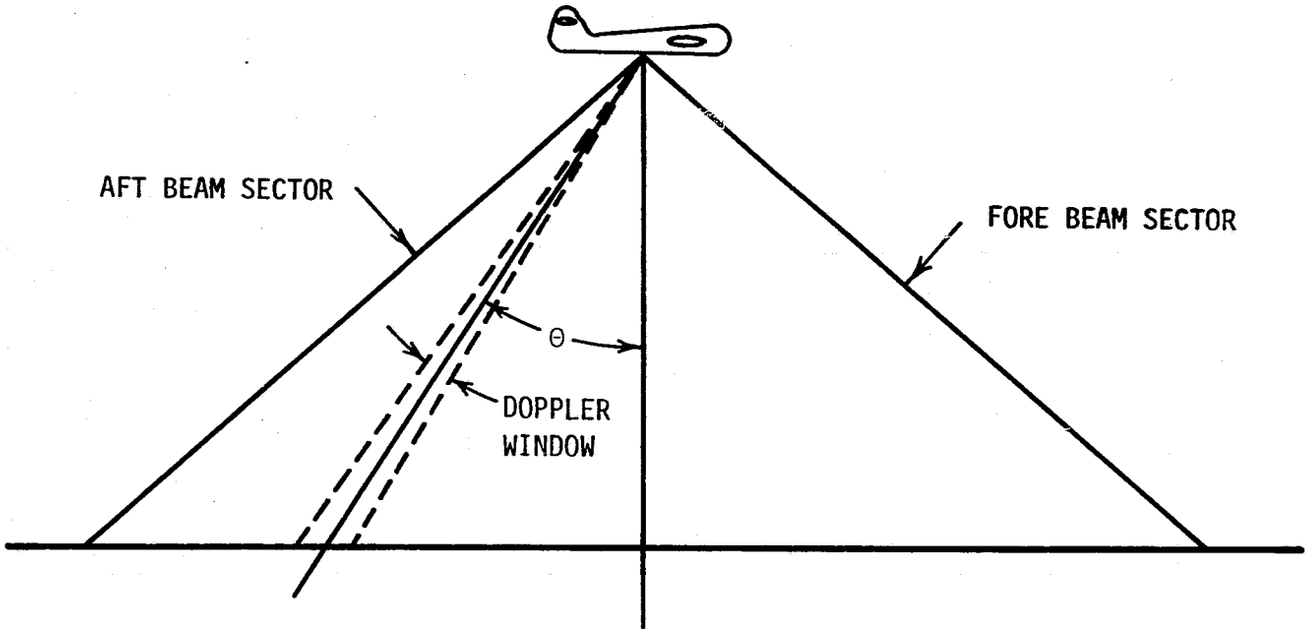


FIGURE 3.1 DOPPLER PROCESSING AND THE FAN BEAM SCATTEROMETER

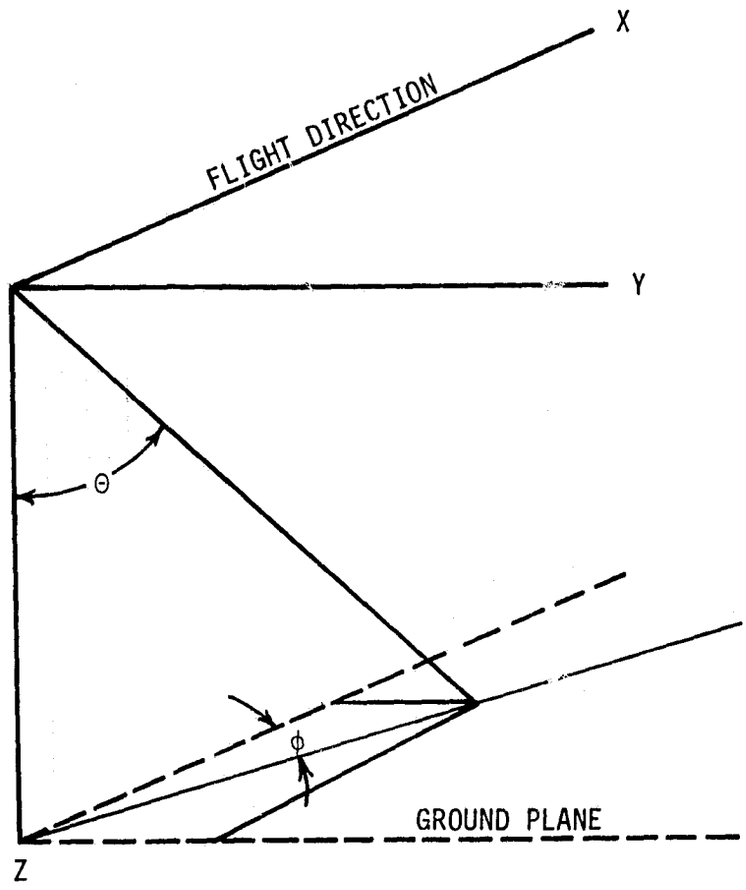


FIGURE 3.2 SCATTEROMETER GEOMETRY

where λ = radar wavelength

G_{tp} = transmit pattern directivity for polarization state p

G_{rq} = receive pattern directivity for polarization state q

W_t = total transmitted power

$R_i = h/\cos\theta_i$

h = aircraft altitude

p,q = indices denoting the transmit and receive polarization states, respectively

In order to primarily discriminate the backscatter within an angular sector of the incident angle θ, a Doppler filter, whose normalized transfer function is given by $H(\omega)$, is employed (see Figure 3.1). The portion of the return power appearing at the output of the filter is therefore given by

$$W_r = \frac{\lambda^2 W_t}{(4\pi)^3} \sum_{i=1}^{N} \sum_{j=1}^{M} G_{tp}(\theta_i,\phi_j)\, G_{rq}(\theta_i,\phi_j)\, \sigma_{pq}(\theta_i,\phi_j) \left|H(\omega_{ij})\right|^2 / R_i^4 \tag{3.2}$$

where

$$\omega_{ij} = 4\pi\, v\, \sin\theta_i\, \cos\theta_j/\lambda \tag{3.3}$$

is the radian frequency associated with the patch in direction (θ_i,ϕ_j). When the normalized scattering coefficient $\sigma^o_{pq}(\theta,\phi)$ is introduced, the double summation may be replace with a double integral given by

$$W_r = \frac{\lambda^2 W_t}{(4\pi)^3} \iint \frac{G_{tp} G_{rq}\, \sigma^o_{pq} \left|H(\omega)\right|^2\, dA}{R^4} \tag{3.4}$$

In the interpretation of W_r it is important to realize that $\left|H(\omega)\right|^2$

participates within the integration since ω is dependent on (θ,ϕ). In an ideal fan beam scatterometer, ϕ ranges over a small interval about zero since the crosstrack antenna beam width is small. As a consequence

$$\omega \simeq 4\pi \, v \sin \theta/\lambda \tag{3.5}$$

It is then observed that $|H(\omega)|^2$ plays the role of a normalized antenna pattern having discriminatory power in the θ dimension.

When the bandwidth of the Doppler filter is sufficiently narrow, σ^o_{pq} may be regarded as constant over the area A spanned by the Doppler bandwidth and the crosstrack beamwidth. The scatterometer equation (3.4) then reduces to the form

$$W_r(\theta_o) \cong \frac{\lambda^2 W_t}{(4\pi)^3} \frac{\sigma^o{}_{pq}(\theta_o)}{h^4} \iint G_{tp} G_{rq} \left|H\right|^2 \cos^4\theta \, dA \tag{3.6}$$

where θ_o is the incident angle corresponding to the center frequency ω_o of the Doppler filter, i.e.,

$$\omega_o = 4\pi \, v \sin \theta_o/\lambda \tag{3.7}$$

The recovery of $\sigma(\theta_o)$ is, therefore, dependent on measurement of W_r, W_t and h and on an estimate of the double integral. In this regard the integral is often approximated by introducing an effective area A_{eff} so that

$$\iint G_t G_r \left|H\right|^2 \cos^4 \theta \, dA = G_t(\theta_o,0) G_r(\theta_o,0) \cos^4 \theta_o \, A_{eff} \tag{3.8}$$

The factor A_{eff} is often formulated on the basis of computed experience.

3.3 Precision in Estimating σ^o_{pq}

In order to estimate σ^o_{pq} at a set of incident angles θ_{oi}, i=1,2,..., n, a bank of filters is required. The center frequency of each filter is chosen in accord with equation (3.7). If $S(\omega)$ is the spectral density of the return signal, then the output of the ith filter H_i is given by

$$W_r(\theta_{oi}) = 2 \int_0^\infty |H_i(\omega)|^2 \quad S(\omega)df \tag{3.9}$$

An alternative method could, instead, measure (estimate) the power spectral density (PSD) of the return signal and then form the return power through an integration, viz.,

$$W_r = 2 \int_{f_{li}}^{f_{ui}} S(\omega)df \tag{3.10}$$

where f_{ui} and f_{li} are the upper and lower corner frequencies associated with the ith angle. In the latter case the measurement of σ^o_{pq} has been reduced to a problem in estimating the power spectral density.

Regardless of the approach, it is necessary to reduce the variance in the estimate of the mean power return W_r to assure a good estimate of the <u>average</u> scattering coefficient σ^o_{pq}. As is well known, the radar return is characterized by heavy fading since the signal has a Rayleigh-like amplitude distribution [2]. As a consequence, it is difficult to estimate the mean squared statistic of such a signal. The theory for the precision in the estimating the mean squared statistic appears in references [2], [3] and others. The extrapolation of this theory to the case where the PSD is to be estimated can be established on an intutive basis and is made precise in reference [4]. The PSD is estimated from the periodogram and is

defined by

$$S_N(k) = \frac{1}{N} \left| \sum_{n=0}^{N-1} s(n)\, e^{-j\, 2\pi\, kn/N} \right|^2 \qquad (3.11)$$

for the kth spectral line for a signal represented in sampled form $\{s(o), s(1),\ldots,s(N-1)\}$.

In the case where analog filtering, detection and integration is performed, it is well known that the standard deviation σ in the estimate of W_r is given by [3]

$$\sigma = W_r/\sqrt{BT} \qquad (3.12)$$

where B is the pre-detection (effective) bandwidth and T is the integration period. The dependence of the variance ratio σ^2/W_r^2 on the BT product is illustrated in Figure 3.3.

Although the variance reduction is a good indication of the improvement in the estimate of W_r, the precision of a system is better conveyed by a statistical 90% confidence interval. The 90% confidence interval for a BT product of 10 or better can be approximated by [5]

$$\overline{W}_r - 1.645\sigma \leq \overline{W}_r \leq \overline{W}_r + 1.645\sigma \qquad (3.13)$$

where $\overline{W}_r$ is the estimate of W_r. When the span of this confidence interval is expressed in dB, it can be written as

$$R = 10\ \text{Log}\ \frac{1 + 1.645/\sqrt{BT}}{1 - 1.645/\sqrt{BT}} \qquad (3.14)$$

The dependence of this precision factor on the time bandwidth product is illustrated in Figure 3.4. It is observed that a $\pm$ 1 dB confidence interval

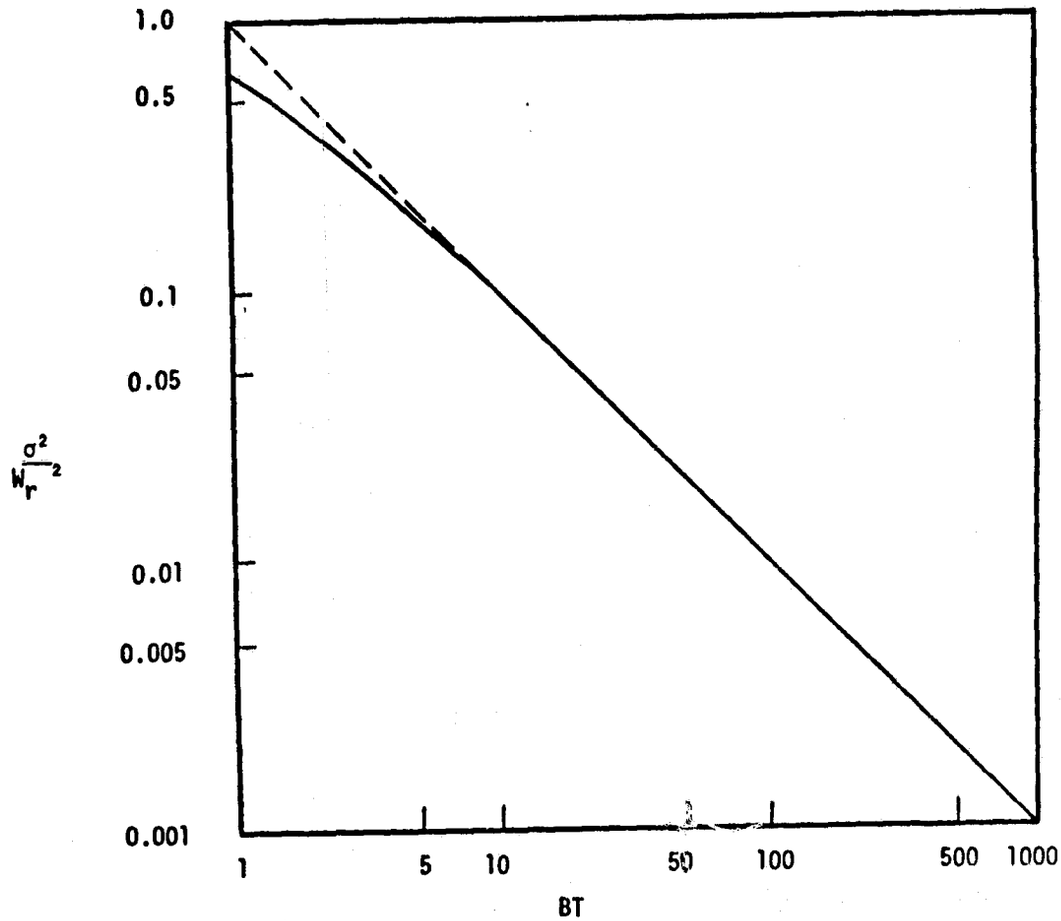


FIGURE 3.3 THE VARIANCE RATIO AS A FUNCTION OF INTEGRATION TIME-BANDWIDTH PRODUCT (FROM REFERENCE [2])

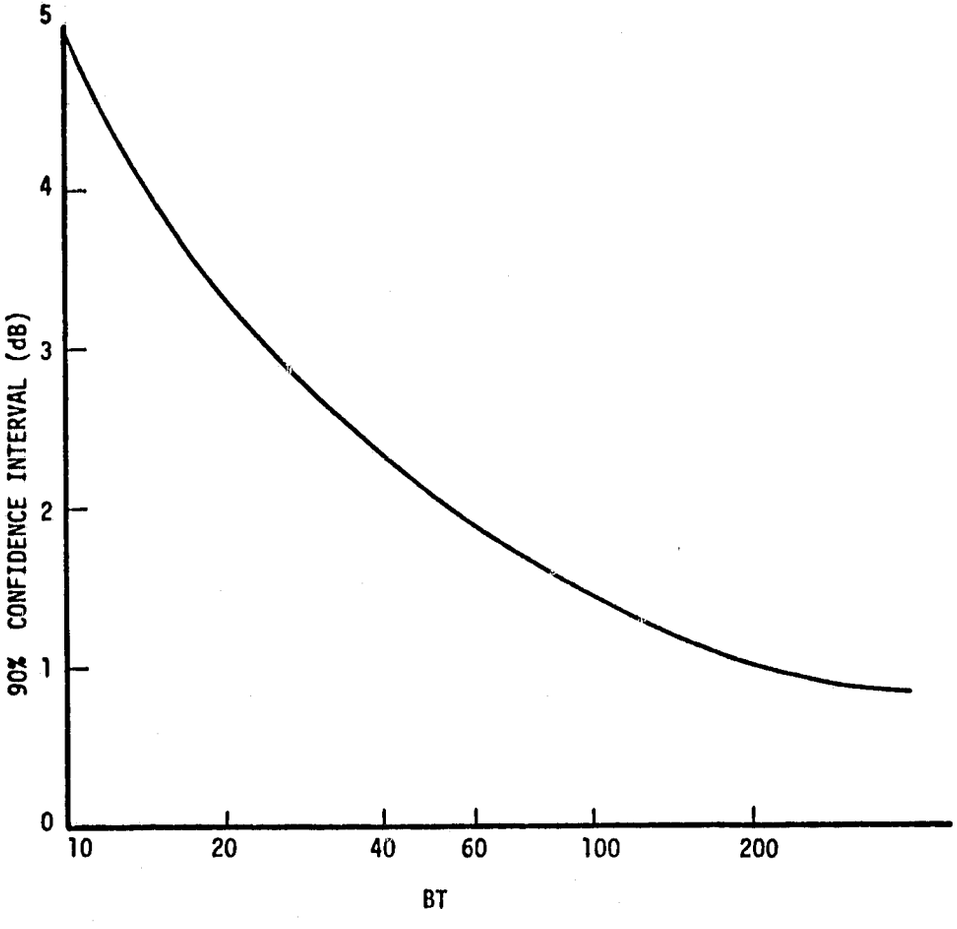


FIGURE 3.4 THE IMPROVEMENT IN PRECISION WITH THE TIME BANDWIDTH PRODUCT

requires a BT product of 50. Furthermore, it is noted that the improvement in precision becomes less rapid when BT > 100.

The above theory can also be applied to the case where the PSD is determined from the periodogram. To obtain precision in the estimate of the PSD, the periodogram must be averaged. This can be accomplished by averaging spectral estimates from a sequence of records and by smoothing adjacent spectral lines from a single record. To make the association between the precision for an analog processor with that for a PSD processor, it should be noted that the PSD estimates are based on the discrete Fourier transform (DFT) of the return signal s(t) (see equation (3.18)). The effective bandwidth associated with DFT processing (filtering) is given by

$$\Delta f = \frac{1}{T} \tag{3.15}$$

where T is the duration of the signal record. It has been assumed that the record is unweighted. Therefore, the time bandwidth product associated with a single line from a single record is simply

$$BT = \Delta f T \tag{3.16}$$

or 1. Since the spectral estimates from a single record are poorly correlated, the BT product can be enhanced by averaging over a window of adjacent spectral lines. The BT product can be further enhanced by averaging line estimates from a sequence of non-overlapping records. So, if N_f adjacent lines from each record are smoothed and N_R non-overlapping records are processed, the resulting BT product for a filter of

width $N_F\Delta f$ is given by

$$BT = (N_F\Delta f)(N_R T) \tag{3.17}$$

or simply $N_F N_R$. This result indicates that the precision improvement is identical for analog and PSD processing.

3.4 Power Spectral Estimation Using the Chirp Z-Transform

As indicated above, the estimate of the PSD can be based on the DFT of a radar return record. If s(n), n=0,1,2,...,N-1 is an N point sequence representing the return signal s(t) over a time interval T, then the DFT of s(t) is defined as

$$F(k) = \sum_{n=0}^{N-1} s(n)\, e^{-j2\pi kn/N} \tag{3.18}$$

where $k \in \{0,1,2,\ldots,N-1\}$. F(k) is interpreted as the spectral amplitude of s(t) at a frequency of k/T when $0 \le k \le \frac{N}{2}$ and at a frequency of -(N-k)/ T for $\frac{N}{2} < k < N - 1$. It has been assumed that N is even. The DFT formulation may be modified through the use of the identity

$$2nk = n^2 + k^2 - (k - n)^2 \tag{3.19}$$

to permit an implementation of the DFT by hardware. The identity results in a DFT given by

$$F(k) = e^{-j\pi k^2/N} \sum_{n=0}^{N-1} s(n) e^{-j\pi n^2/N}\, e^{\,j\pi(k-n)^2/N} \tag{3.20}$$

The implication of the above result is that s(n) must be first down-chirped with $e^{-j\pi n^2/N}$, convolved with an up-chirp $e^{j\pi n^2/N}$ and then post multiplied by a down-chirp $e^{-j\pi k^2/N}$. The pre and post multiplications may be

performed by analog methods and the convolution may be formed with transversal filters using charge coupled devices. The transversal filter requires 2N-1 stages to implement the DFT. When forming the PSD, the post chirp may be discarded since it does not affect the amplitude.

A more efficient means for implementing the transversal filter, requiring only N stages in the transversal filter, is based on the sliding DFT. The sliding transform is defined as

$$F_s(k) = \sum_{n=k}^{k+N-1} s(n)e^{-j2\pi nk/N} \tag{3.21}$$

and differs from the non-sliding version in that the input sequence is shifted forward one sample for each new spectral estimate. The transform consequently operates continuously. Spectral estimates on the same line are updated every N samples since $e^{-j\pi nk/N}$ is modulo N in the parameter k. As a result of the sliding action, phase information is destroyed; nevertheless, the magnitude information important to the PSD estimation is preserved.

It can be shown that through the use of the identity of equation (3.19), the sliding transform can be rewritten as

$$F_s(k) = e^{-j\pi k^2/N} \sum_{m=1}^{N} e^{j\pi(m-N)^2} s(k - m + N)e^{-j\pi(k - m + N)^2/N} \tag{3.22}$$

This result implies that the input must be pre-chirped by a factor $e^{-j\pi(m-N)^2/N}$, convolved with $e^{j\pi(m-N)^2/N}$ and then post multiplied with $e^{-j\pi k^2/N}$ to form the sliding DFT. When the PSD is required, the post

multiplication may be replaced with a squaring operation to yield $|F_s(k)|^2$. Since the pre-multiplication is periodic in m, the CZT filter can operate on the input signal continuously with the transversal filter only requiring N stages.

3.5 The Application of the CZT to Doppler Filtering

3.5.1 Signal Processing Theory

A simplified block diagram of the CW fan beam scatterometer is illustrated in Figure 3.5. The transmitter illuminates the terrain at a radian frequency of ω_o. The backscattered signal arriving at the receiver may be denoted as

$$s(t) = a(t) \cos [\omega_o t + \phi(t)] \tag{3.23}$$

where $a(t)$ and $\phi(t)$ may be regarded as random variables. The spectrum of $s(t)$ is depicted symbolically in Figure 3.6a. The fore and aft spectra are distinguished from one another by "coloring" the fore spectrum as a rectangle and the aft spectrum as a triangle. As indicated in the receiver chain, the return signal is split equally into two channels. This signal in the upper channel is coherently demodulated with $\cos \omega_o t$ and low pass filtered to yield

$$x(t) = \tfrac{1}{2} a(t) \cos\phi(t) \tag{3.24}$$

The upper channel is commonly called the cosine channel or the in-phase (I) channel. Demodulation and low pass filtering in the lower channel yields

$$y(t) = \tfrac{1}{2} a(t) \sin\phi(t) \tag{3.25}$$

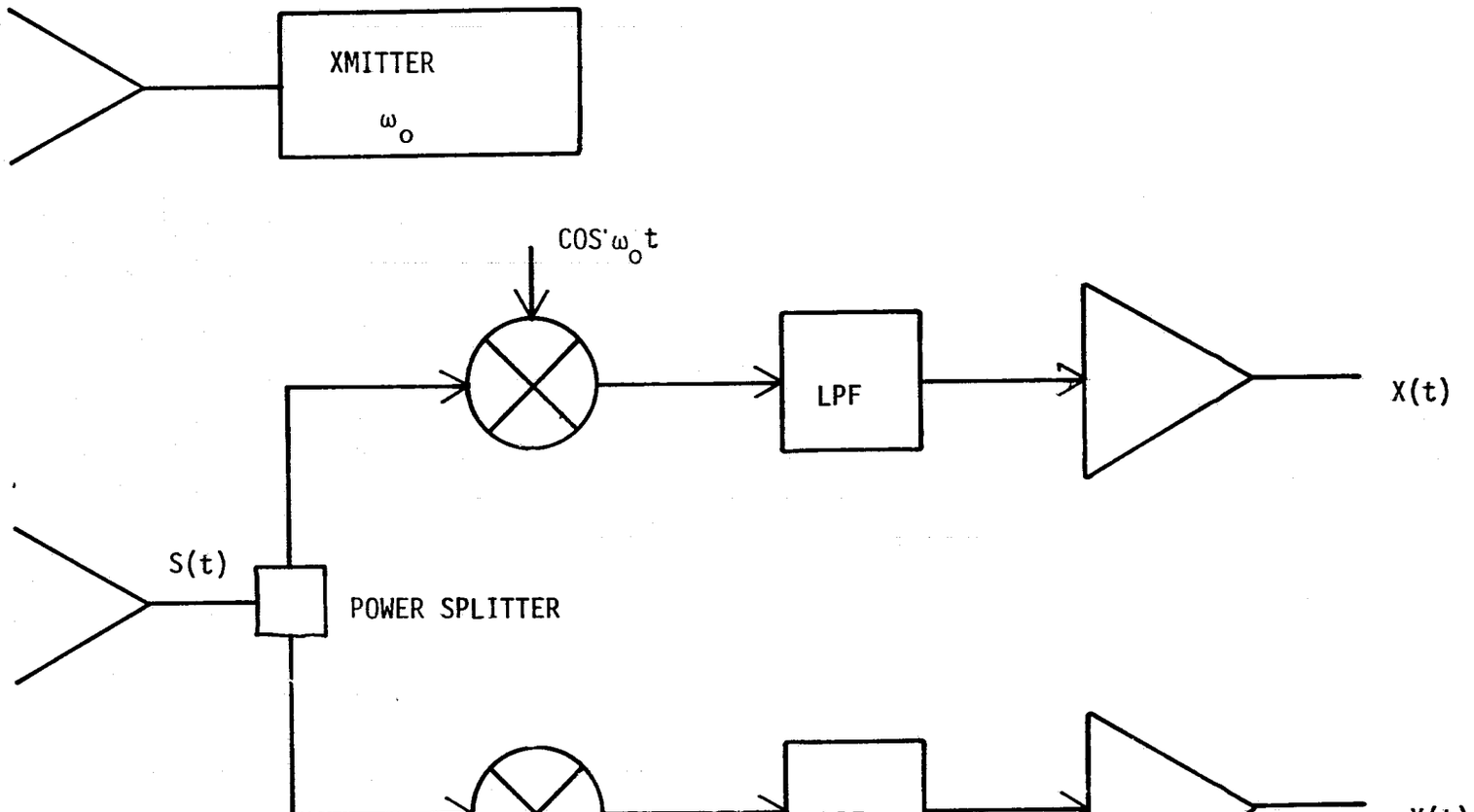


FIGURE 3.5 A SIMPLIFIED BLOCK DIAGRAM OF THE CW FAN BEAM SCATTEROMETER

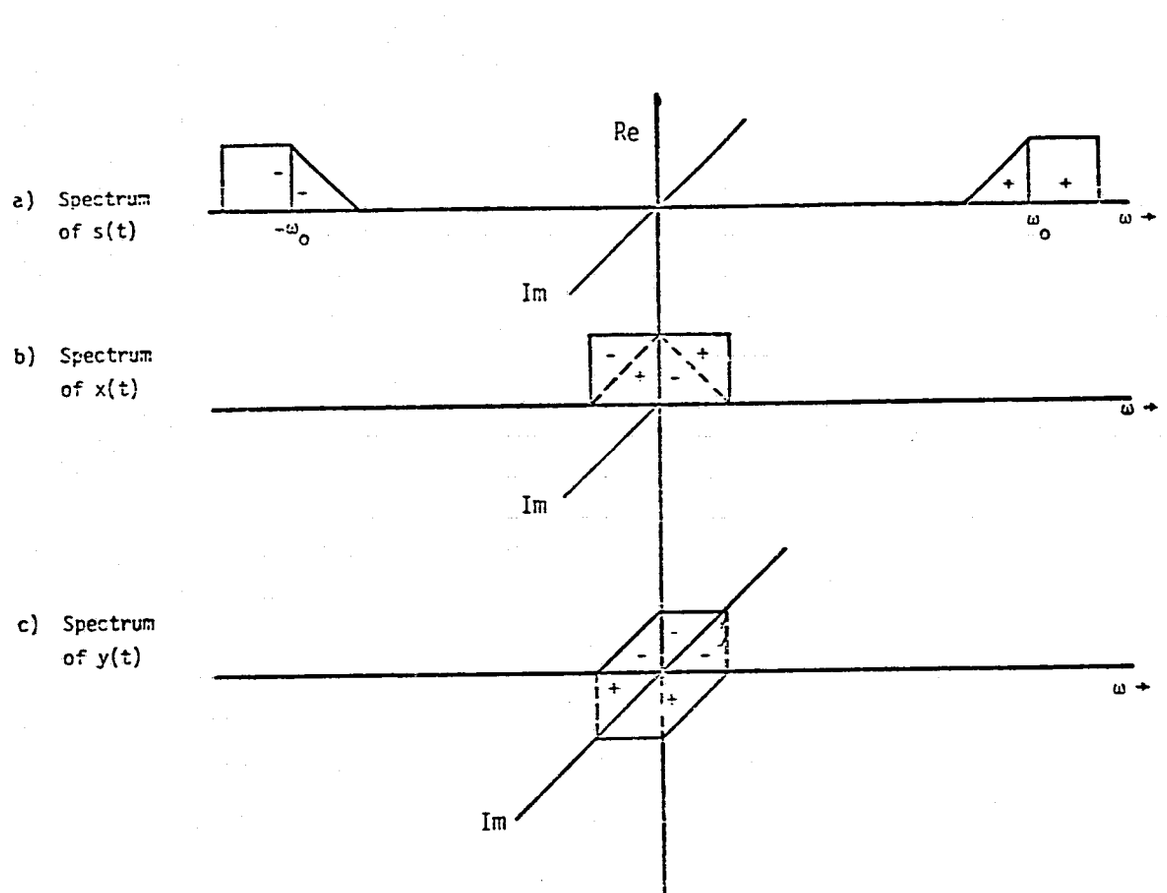


FIGURE 3.6 SPECTRA OF INTEREST

where product modulation with the quadrature reference sin $\omega_o t$ has occurred. The lower channel is commonly called the sine channel or the quadrature (Q) channel. The spectra of x(t) and y(t) are illustrated in Figures 3.6b and 3.6c, respectively.

A comparison of Figures 3.6b and 3.6c shows that the aft and fore spectra can be retrieved simultaneously if a complex signal

$$z(t) = x(t) + j\, y(t) \tag{3.26}$$

is formed. An examination of the spectrum of z(t) indicates that the aft spectrum occurs for $\omega > 0$ and the fore spectrum for $\omega < 0$.

Since the DFT can also be applied to complex signals as well as real signals, the above result shows that the fore and aft spectra can be simultaneously filtered using CZT techniques to implement the DFT. The configuration for implementing the CZT using charge coupled devices and analog multipliers is described below.

3.5.2 The Implementation Technique

It is advantageous to use the sliding version of the CZT, since many sequential measurements are required to estimate the PSD. The actual implementation method is best understood by separating the real and imaginary parts of the sliding CZT. When the sliding CZT of z(t) is taken, $Z_s(k)$ can be rewritten in the form

$$\begin{aligned} |Z_s(k)| = \sum_{m=1}^{N} & [\cos\pi(m-N)^2/N + j \sin\pi (m-N)^2/N] \\ & [x(k-m+N) + jy(k-m+N)]. \\ & [\cos\pi(k-m+N)^2/N - j \sin\pi(k-m+N)^2/N] \end{aligned} \tag{3.27}$$

A careful interpretation of the arguments within the above magnitude suggests the implementation technique shown in Figure 3.7. Both x(t) and y(t) are pre-chirped and appropriately summed to form the real and imaginary valued entries into the transversal filter bank. Four transversal filters are required to form the cross multiplication products of the complex valued signal entering the filters. The real and imaginary parts of the CZT without the post-multiplication are formed by differencing the outputs of the transversal filters as illustrated in Figure 3.7.

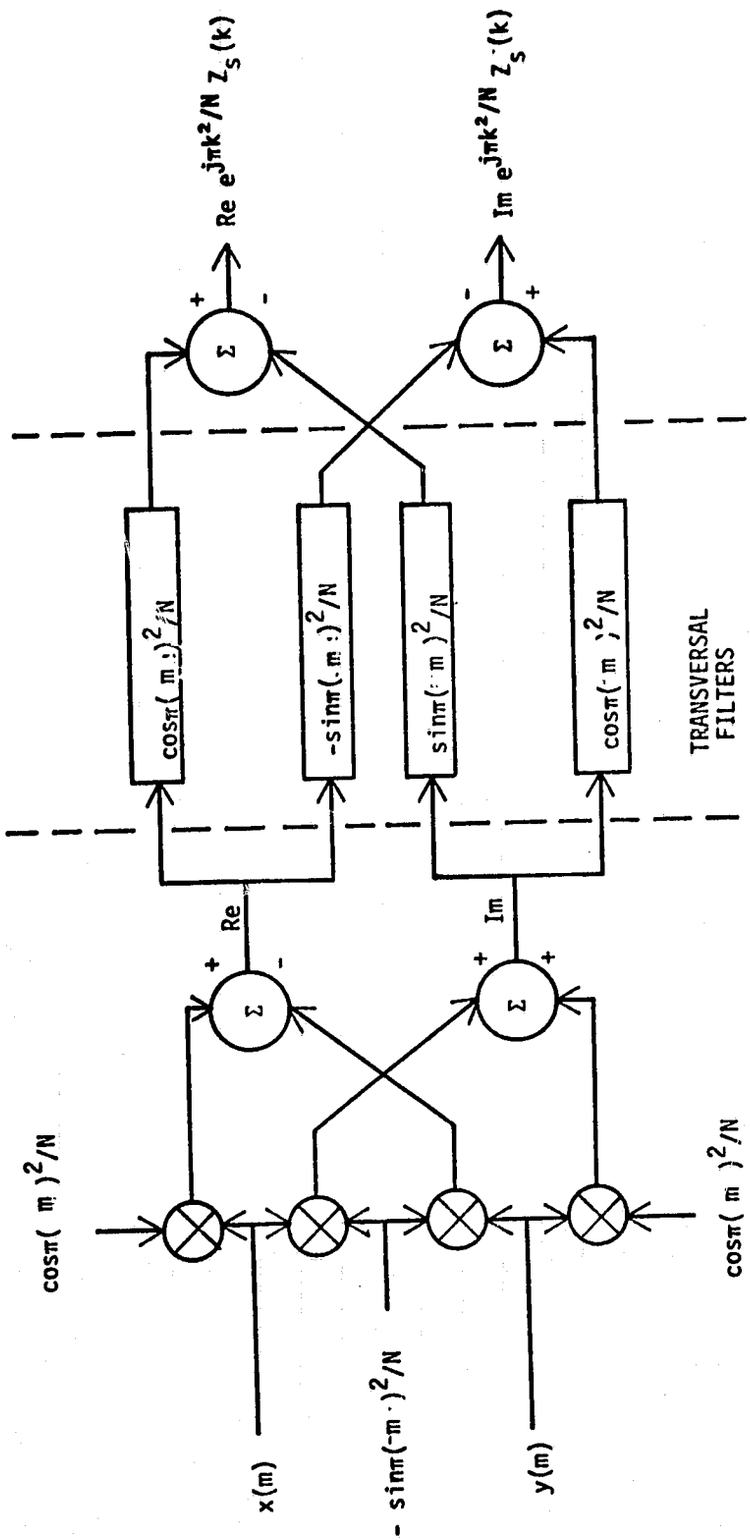


FIGURE 3.7 A TECHNIQUE FOR IMPLEMENTING THE CZT WITHOUT POST MULTIPLICATION

4.0 SYSTEM DESIGN RATIONALE

4.1 Introduction

As is the case with any system, the actual design is a compromise between user requirements and system constraints. The design of real time processors for the NASA scatterometers is no exception. Ideally the scatterometer should provide high precision estimates of σ° with infinitesimal angular and ground resolutions. However, as will be shown below, the precision, angular resolution and ground resolution interact in such a way to prevent maximizing all three parameters simultaneously. In addition to maximizing the precision and resolution parameters, the user is also interested in achieving a reasonable accuracy to permit comparative analysis of the processed data at different view angles and polarizations and with scatterometer data from other sources (also presumably calibrated).

A list of the factors which can potentially influence the performance of the scatterometer/processor system is shown in Table 1. The source of error is described in the left-hand column. The system performance factors most influenced by the error source is reflected in the middle column. The origin of the error within the system is identified in the right-hand columns. Most of the performance parameters are manageable by the processor provided that the scatterometer has been appropriately designed. Those that are manageable are treated below as well as in subsequent chapters to develop the processor design rationale.

TABLE 4.1 FACTORS INFLUENCING SYSTEM PERFORMANCE

Error Source	Affected Performance Factor	Origin in System			
		Scatterometer	Processor	Aircraft	Target
1. Fading Signal	Precision				X
2. Finite Doppler Bandwidth	Precision & Angular Resolution		X		
3. Finite Record Length	Ground Resoltuion & Precision		X		
4. Filter Sidelobe Level	Accuracy		X		
5. Bit Truncation	Precision		X		
6. Inversion Approximation	Accuracy		X		
7. Uncertainty in Altitude	Accuracy			X	
8. Aircraft Attitude a) Illuminated area b) Polarization decomposition	Accuracy			X X	
9. Transmitted Power	Accuracy	X			
10. Polarization	Accuracy	X			
11. Non-Stationary Return	Accuracy				X
12. Beamwidth	Accuracy & Angular Resolution	X			
13. Pattern Sidelobes	Accuracy	X			
14. Pattern Gain	Accuracy	X			

4.2 Definition of System Design Parameters

To identify the processor's mode of operation it is important to define various parameters associated with fan beam systems. In this regard such terms as angular resolution, ground resolution, scan length, beam resolution, ground track coverage, etc. must be clarified to arrive at the impact of these parameters on the system design.

Angular Resolution

As indicated in Section 3.2 the angular resolution of a fan beam scatterometer is dictated by the physical beamwidth in the crosstrack dimension and by the bandwidth of the Doppler filter in the intrack dimension. If $H(\omega)$ denotes the normalized voltage transfer function of the Doppler filter, then an affective bandwidth may be defined as

$$B = \frac{1}{2\pi} \int_{-\infty}^{\infty} |H(\omega)|^2 \, \omega \tag{4.1}$$

where the normalization has been applied so that $\max_{\omega} \{| H(\omega) |\} = 1$ when $H(\omega)$ represented in low pass form. An effective intrack beamwidth may be related to the effective bandwidth through the Doppler relationship

$$\Delta\theta = \frac{\lambda B}{2v\cos\theta} \tag{4.2}$$

when $\phi \simeq 0$. $\Delta\theta$ is defined to be the angular resolution.

Beam Resolution

The beam resolution in the cross track dimension is given by

$$\rho_c = (h \tan\theta)\, \Delta\phi \tag{4.3}$$

where $\Delta\phi$ is the crosstrack angle subtended by the two way beam when projected on the ground plane. The beam resolution in the intrack dimension is

$$\rho_B = \frac{h}{\cos^2\theta} \Delta\theta \tag{4.4}$$

See Figure 4.1 to clarify these definitions.

Scan Length

Scan length is simply the ground track distance vT traversed by the aircraft during a single integration period T where v is the ground velocity. See Figure 4.2.

Ground Track Coverage

The ground track coverage L_c is that entire length over which a radar return was observed. The coverage includes the scan length as well as the initial coverage within the beam, ie.,

$$L_c = \rho_B + vT$$

See Figure 4.2.

Ground Resolution

The ground resolution may be defined in several ways. However, for the purposes of this design effort the ground resolution is defined as an effective ground length over which radar returns have primarily contributed to the measurement as implied in Figure 4.2. It consequently emphasizes that portion of the ground track which is repeatedly in view within the beam subtended by $\Delta\theta$.

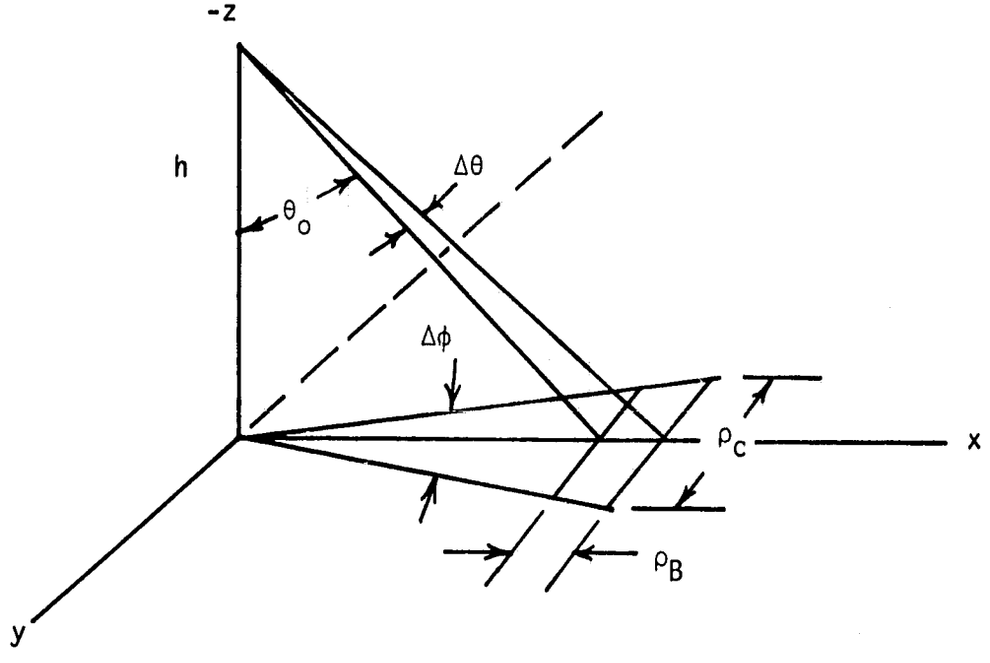


FIGURE 4.1 VARIOUS RESOLUTION PARAMETERS

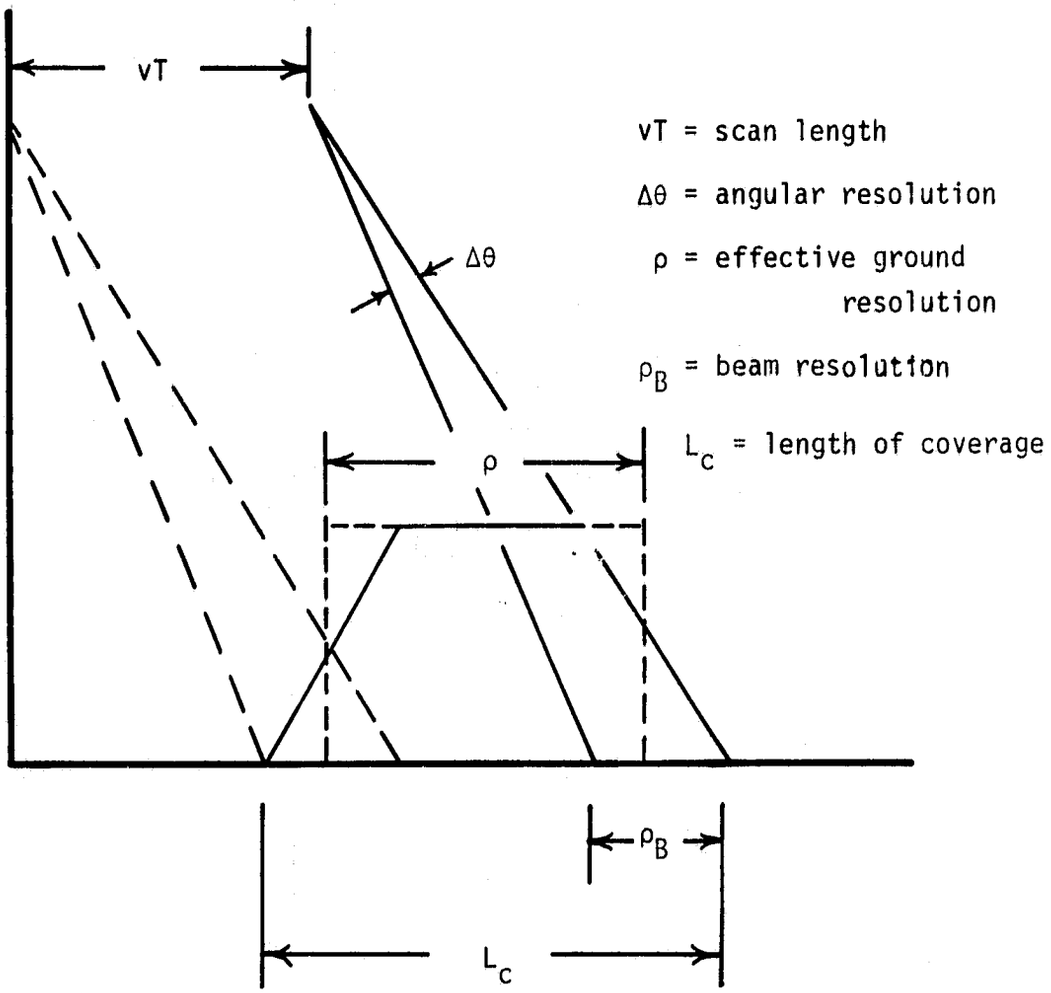


FIGURE 4.2 VARIOUS CELL PARAMETERS

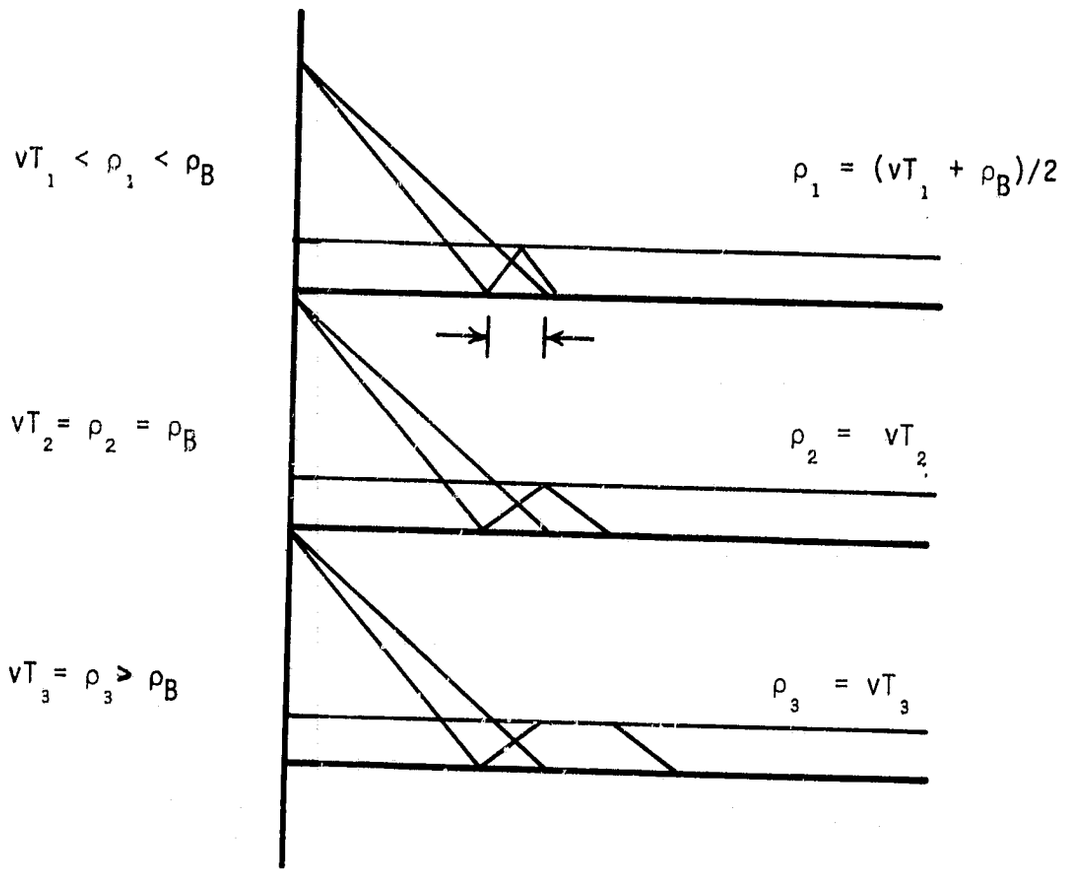


FIGURE 4.3 COMPARISON OF THE RESOLUTION PARAMETERS WHEN (a) $vT_1 < \rho_B$, (b) $vT_2 = \rho_B$ and (c) $vT_3 > \rho_B$

Three cases may be identified as illustrated in Figure 4.3. The ground resolution is therefore defined as

$$\rho = \begin{cases} \dfrac{vT + \rho_B}{2} & \text{if } vT < \rho_B \\ \rho_B & \text{if } vT = \rho_B \\ vT & \text{if } vT > \rho_B \end{cases} \tag{4.5}$$

From the above definitions an important observation can be withdrawn. When the three ground resolution cases are ordered as implied in Figure 4.3 it may be shown that

$$\frac{T_2}{T_1} \geq \frac{\rho_2}{\rho_1} \tag{4.6}$$

and

$$\frac{T_2}{T_3} = \frac{\rho_2}{\rho_3} \tag{4.7}$$

where ρ_i is the ground resolution corresponding to T_i. For a constant Doppler bandwidth these results imply the following conclusion:

Case 2 maximizes the ground resolution consistent with maximizing the precision. It is also interesting to note that Case 2 maximizes the precision for a given coverage interval L_c as demonstrated in Appendix A. In this case the design criterion requires that

$$\rho_B = vT \tag{4.8}$$

at any incident angle. This conclusion will be used in identifying appropriate design theories below.

4.3 The Theory for Constant Precision, Constant Angular Resolution and Constant Ground Resolution Designs

Among the many system designs which could be considered it is helpful to limit considerations to three basic design approaches: (1) constant precision, (2) constant angular resolution and (3) constant ground resolution. The theory for each is presented below and their characteristics are compared in a final subsection.

4.3.1 A Constant Precision Design

A constant precision design approach requires a constant BT product at each incident angle to be processed. The implication deduced from Section 4.2 is helpful in assigning B and T so as to achieve an acceptable BT product. At the smallest incident angle θ_1 the BT product may be maximized for a given coverage L_c by requiring

$$\rho_{B^1} = vT \tag{4.9}$$

Then

$$T = \frac{h\,\Delta\theta_1}{v\,\cos^2\theta_1} \tag{4.10}$$

and

$$B = 2v\,\cos\theta_1\,\Delta\theta_1/\lambda \tag{4.11}$$

The viewing window at θ_1 is therefore specified as

$$\Delta\theta_1 = \sqrt{\frac{BT\,\lambda\,\cos^2\theta_1}{2h}} \tag{4.12}$$

where BT is chosen to achieve the desired precision. When $\Delta\theta_1$ is withdrawn from equation (4.12), T and B are uniquely assigned by equations

(4.10) and (4.11), respectively. On pragmatic grounds the integration time must be constant at all viewing angles. Therefore B is also constant. As a result, the angular resolution, beam resolution, and ground resolution at the remaining view angles θ_k become

$$\Delta\theta_k = \frac{\lambda B}{2v\cos\theta_k} \tag{4.13}$$

$$\rho_{Bk} = \frac{\lambda h B}{2v\cos^3\theta_k} \tag{4.14}$$

and

$$\rho_k = \frac{\lambda h B}{4v\cos^3\theta_1} \left\{1 + \frac{\cos^3\theta_1}{\cos^3\theta_k}\right\} \tag{4.15}$$

From the above results it is noted that the angular resolution is inversely proportional to $v\cos\theta_k$, the beam resolution is proportional to $h/v\cos^3\theta_k$ and the ground resolution is proportional to $h\left\{1 + \frac{\cos^3\theta_1}{\cos^3\theta_k}\right\}$

4.3.2 Constant Angular Resolution

A constant angular resolution approach requires that $\Delta\theta_k$ be constant at all viewing angles, say $\Delta\theta$. Once $\Delta\theta$ is assigned, the bandwidth at each θ_k is given by

$$B_k = \frac{2v\cos\theta_k\Delta\theta}{\lambda} \tag{4.16}$$

and the beam resolution by

$$\rho_{Bk} = \frac{h\Delta\theta}{\cos^2\theta_k} \tag{4.17}$$

The precision and ground resolution require a rationale to assign T. Once again it is convenient to maximize BT at the smallest incident angle for a given coverage. This requires that

$$\rho_{B_1} = vT \tag{4.18}$$

The remaining parameters then become

$$T = \frac{h\Delta\theta}{v\cos^2\theta_1} \tag{4.19}$$

$$(BT)_k = \frac{2h\cos\theta_k\,\Delta^2\theta}{\lambda\cos^2\theta_1} \tag{4.20}$$

and

$$\rho_k = \frac{h\Delta\theta}{2\cos^2\theta_1}\left\{1 + \frac{\cos^2\theta_1}{\cos^2\theta_k}\right\} \tag{4.21}$$

at each viewing angle $\theta_k > \theta_1$.

4.3.3 Constant Ground Resolution

A constant ground resolution approach requires that ρ be constant at each incident angle. With ρ specified, the precision may be maximized at each incident angle by requiring $\rho = \rho_B = vT$ in accord with Section 4.2. As a result of this imposition

$$\Delta\theta_k = \rho\cos^2\theta_k/h \tag{4.22}$$

$$T = \rho/v \tag{4.23}$$

and

$$(BT)_k = 2\rho^2 \cos^3\theta_k/\lambda\, h \qquad (4.24)$$

4.4 A Comparison of the Design Approaches

To evaluate the three design approaches, the nominal design guidelines shown in Table 4.2 were employed for C band and L band systems.

The guidelines were primarily applied at the smallest incident angle and were allowed to vary at the larger incident angles depending on the design approach. The results of this evalution are shown graphically in Figures 4.4 and 4.11.

The first four graphs apply to the C band processor whereas the latter four apply to the L band processor.. Each figure identifies and compares a single system performance parameter over the entire range of incident angles for the three design approaches. The graphs are parametrically identified by the design approach: CP = constant precision, CGR = constant ground resolution and CAR = constant angular resolution.

Table 4.2 Nominal Design Guidelines

Parameter	Value	Units
Aircraft velocity	150	knots
Aircraft altitude	1500	feet
Angular resolution (nominal)		
C band	3	degrees
L band	6	degrees
Ground resolution(nominal)		
C band	25	meters
L band	50	meters
Precision factor (nominal)		
C band	50	
L band	50	

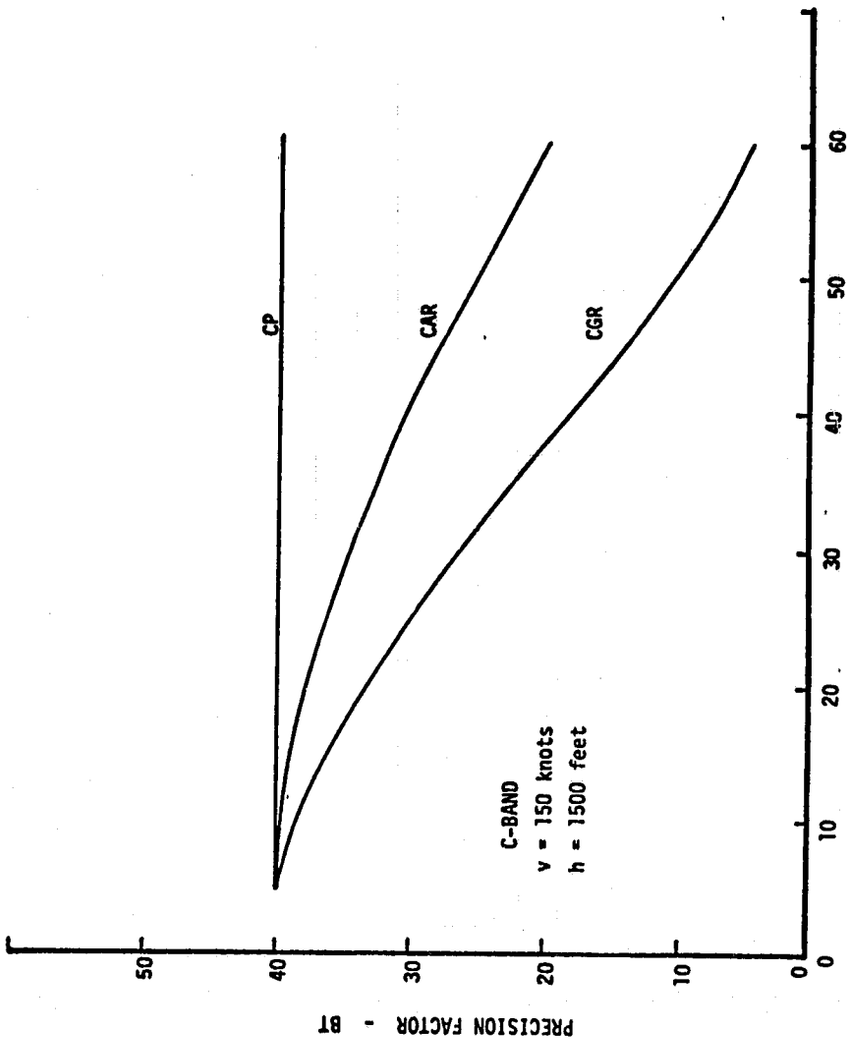


FIGURE 4.4 THE DEPENDENCE OF PROCESSING PRECISION ON VIEW ANGLE FOR THE THREE DESIGN APPROACHES

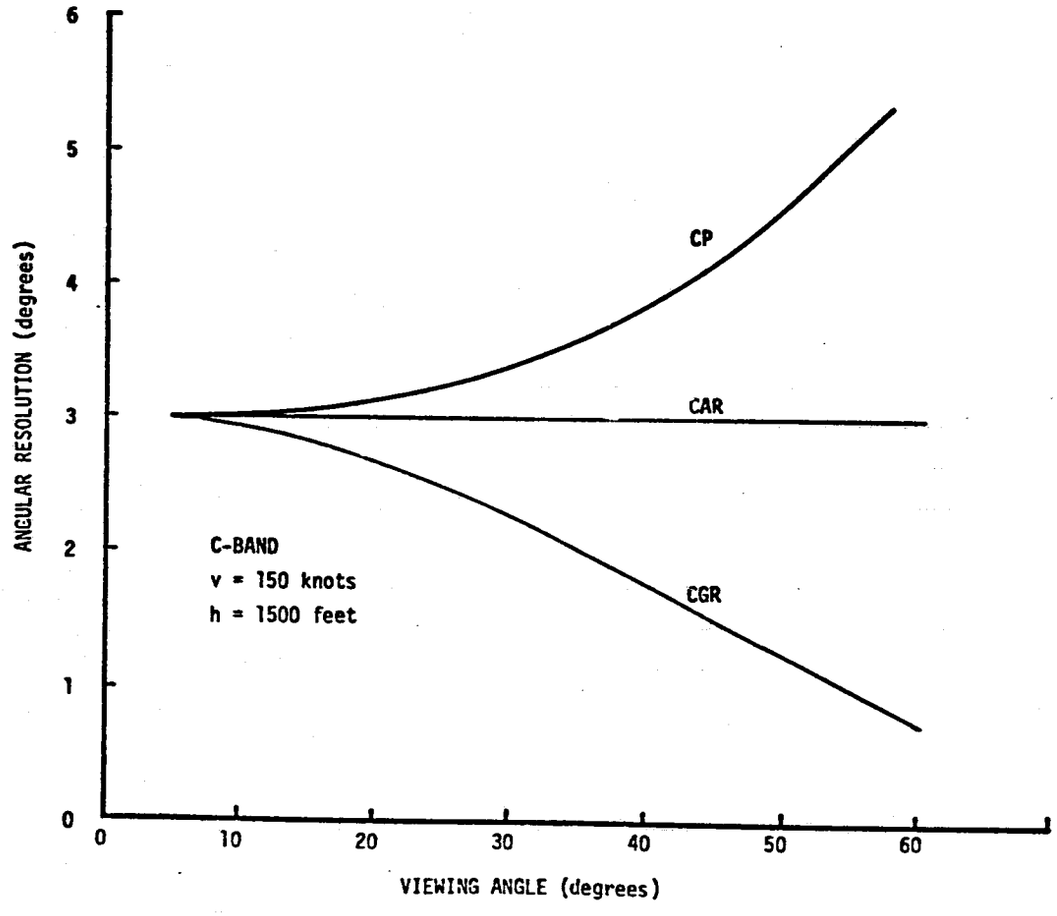


FIGURE 4.5 THE DEPENDENCE OF ANGULAR RESOLUTION ON VIEWING ANGLE FOR THE THREE DESIGN APPROACHES

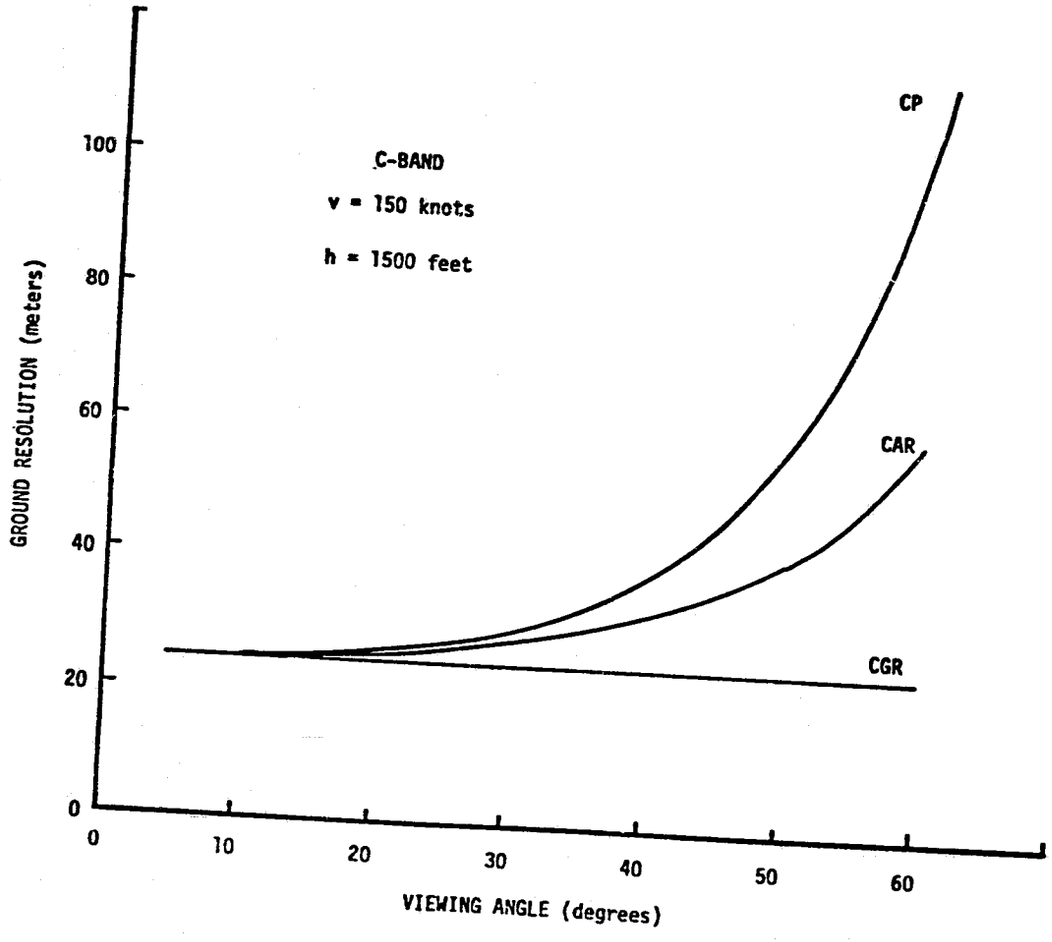


FIGURE 4.6 THE DEPENDENCE OF GROUND RESOLUTION ON VIEWING ANGLE FOR THE THREE DESIGN APPROACHES

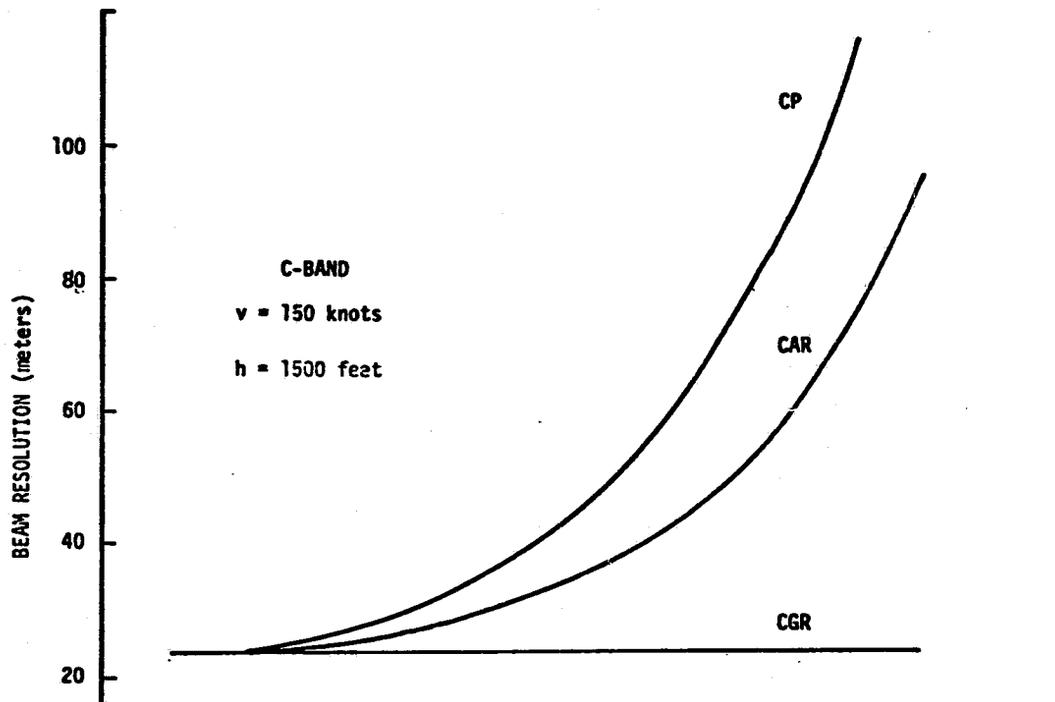


FIGURE 4.7 THE DEPENDENCE OF BEAM RESOLUTION ON VIEWING ANGLE FOR THE THREE DESIGN APPROACHES

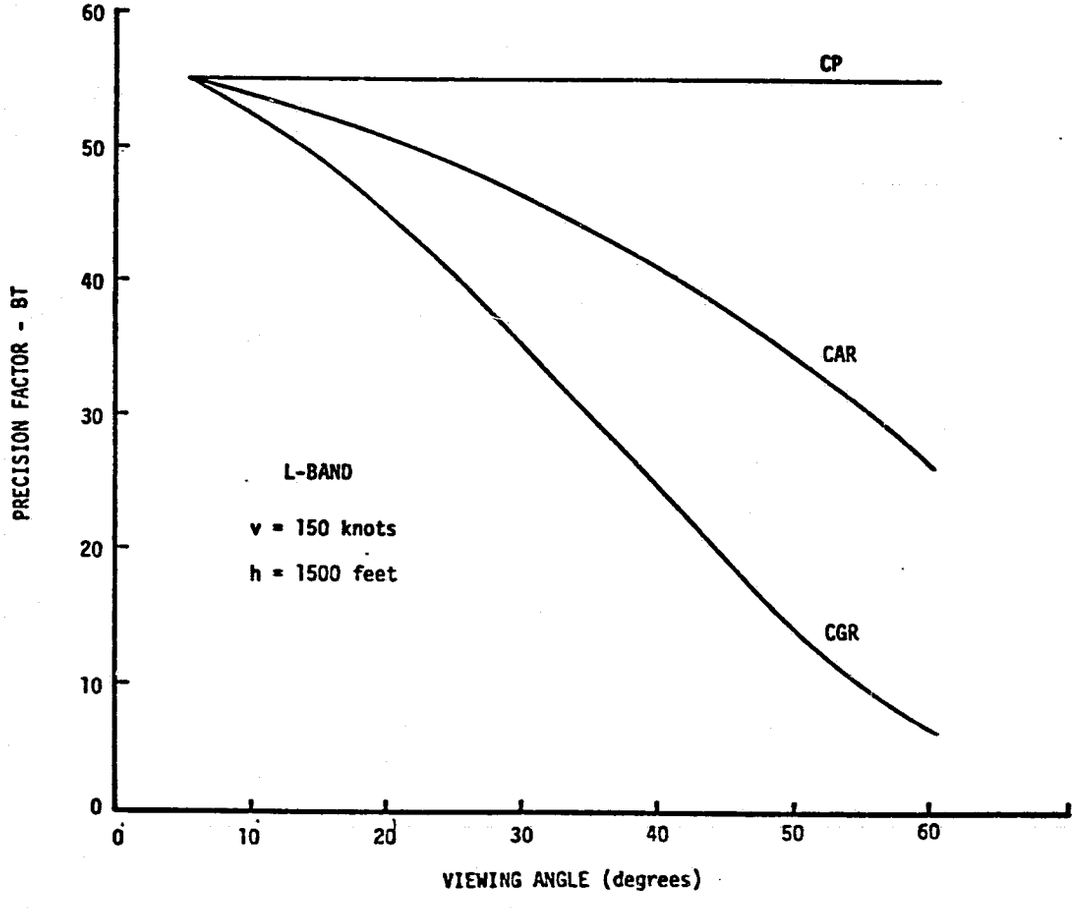


FIGURE 4.8 THE DEPENDENCE OF PROCESSING PRECISION ON VIEW ANGLE FOR THE THREE DESIGN APPROACHES

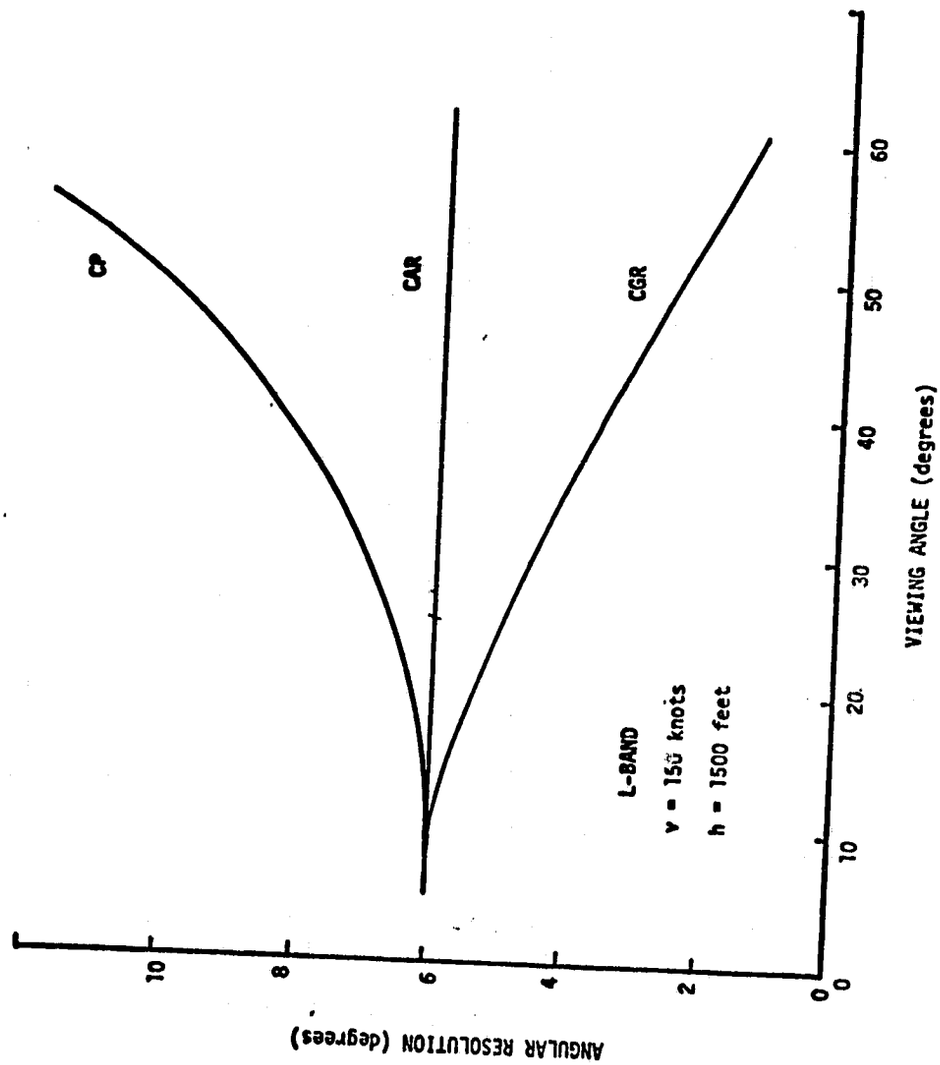


FIGURE 4.9 THE DEPENDENCE OF ANGULAR RESOLUTION ON VIEWING ANGLE FOR THE THREE DESIGN APPROACHES

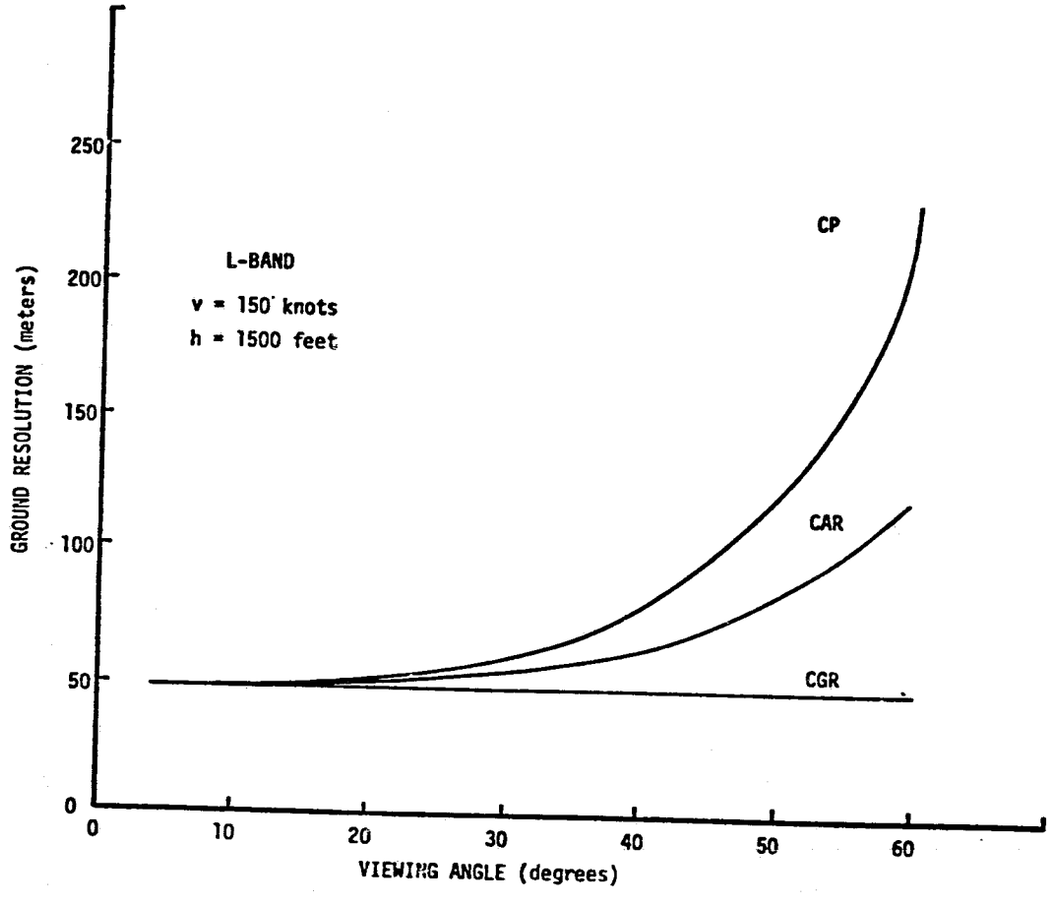


FIGURE 4.10 THE DEPENDENCE OF THE GROUND RESOLUTION ON VIEWING ANGLE FOR THE THREE DESIGN APPROACHES

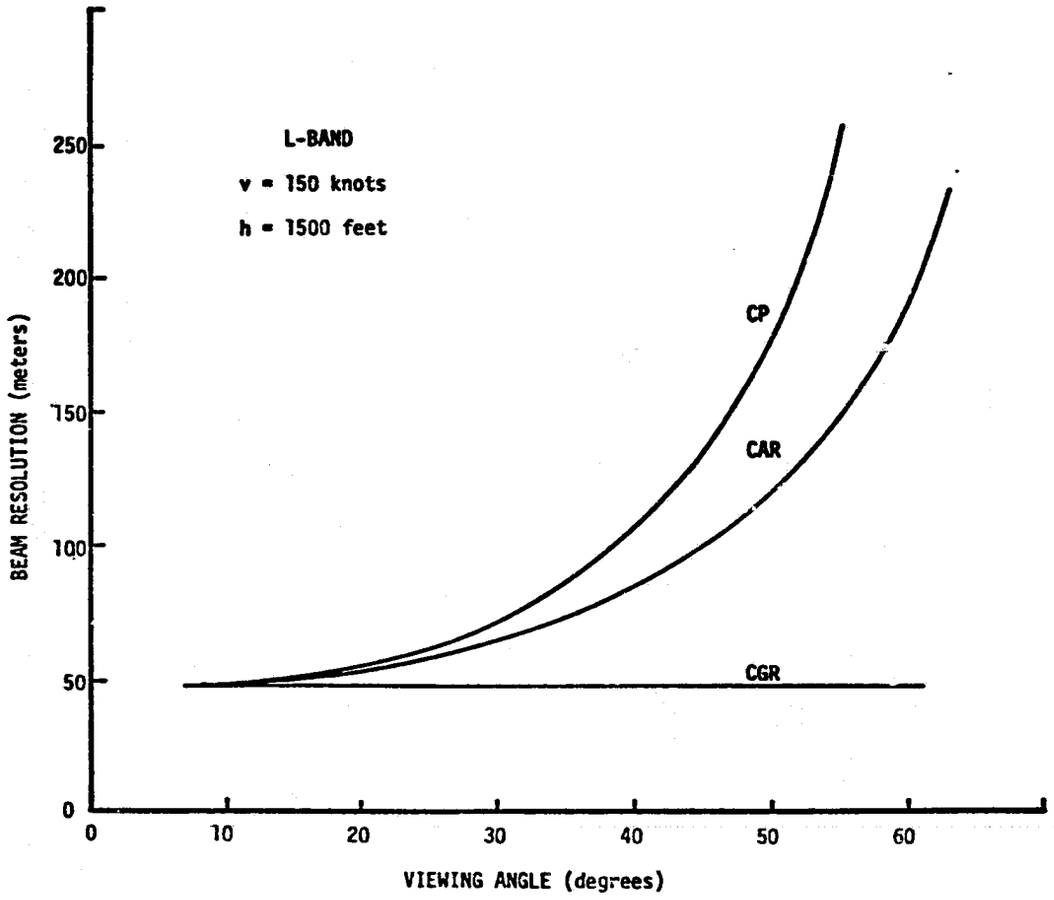


FIGURE 4.11 THE DEPENDENCE OF BEAM RESOLUTION ON VIEWING ANGLE FOR THE THREE DESIGN APPROACHES

From these graphs the following features are noted:

1). When constant precision is imposed, the angular, beam and ground resolutions degrade with incident angle; however, the degradations are only significant for angles greater than 45°.

2). When constant ground resolution is imposed, constant beam resolution is also realized. The precision, however, degrades at the larger incident angles but is useable to 50°. The angular resolution increases rapidly at the large incident angles. The very high angular resolution at the large incident angles may preclude selecting a correct pattern gain at the large incident angles when converting to $\sigma°$.

3). When constant angular resolution is imposed, the beam resolution and ground resolution degrade at large incident angles. The performance, in general, lies between the constant ground resolution and constant precision design approaches.

If a single design approach were to be selected among the three, it is apparent that the constant angular resolution approach represents a good compromise between constant precision and constant ground resolution. The constant precision and constant ground resolution designs may, however, suit some experiments better. The constant precision design is attractive in those cases where high precision is required on a single cell, particularly at the larger viewing angles. The constant precision approach may also be helpful in those cases where a large spatial average is required. The constant ground resolution approach is attractive for those applications where well metered - high resolution data are required along the intrack dimension. This design approach is consequently attrac-

tive for those targets which are highly nonhomogeneous.

Since the scatterometer processor is under software control, it is conceivable to provide an experimenter with any design option. The constant angular resolution design represents a good compromise among the approaches, however, when a single approach must be taken.

5.0 SYSTEM ARCHITECTURE AND OVERVIEW

5.1 Target and Development System Architectures

The objective of the scatterometer processing system is to provide real and post time conversion of two channels of scatterometer data, like and cross polarized signals, into σ^0 estimates at eight (8) viewing angles: 5^0, 10^0, 15^0, 20^0, 30^0, 40^0, 50^0 and 60^0. Processor designs were to be developed for the NASA 4.75 GHz and 1.6 GHz fan beam scatterometers. The efforts were 1) to emphasize a standardize design approach suitable for use with these scatterometers as well as future scatterometer systems and 2) to utilize design experience from the previous 13.3 GHz scatterometer processor project. An appropriate target system architecture using the CZT approach to Doppler processing and meeting the above stated objectives is illustrated in Figure 5.1. The system consists of two major subsystems, viz., the PSD estimation subsystem and the micro-processing subsystem. A number of interface units are also provided to permit control of the system, entry of data and storage of processed data. Also a special alignment generator not required during operation but helpful in aligning the CZT filtering unit prior operation is shown.

The PSD estimation subsystem has two channels to handle like and cross polarized return signals. Each channel converts the quadrature signals into discrete PSD estimates over the fore and aft Doppler spectra simultaneously. The estimation technique is based on the CZT technique which may be regarded as an analog technique of implementing the discrete Fourier transform (DFT). The detection (squaring) and accumulation (averaging) of the spectral amplitudes is performed digitally by a

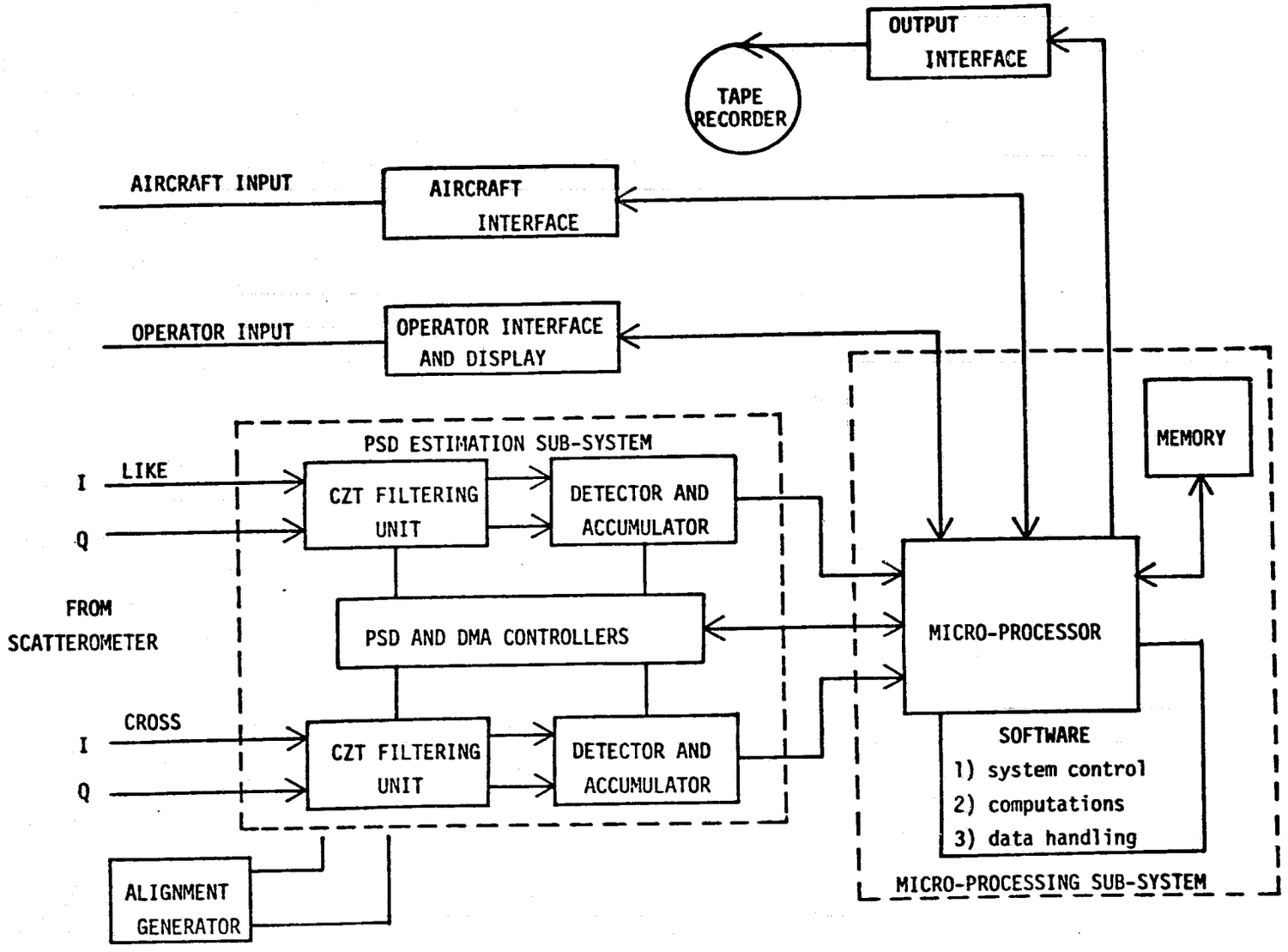


FIGURE 5.1 ARCHITECTURE OF THE TARGET SYSTEM

dedicated processor.

The micro-processing subsystem is composed of a digital micro-computer, memory and associated software. The role of the micro-processing subsystem is 1) to control the scatterometer processor through a software operating system, 2) to select appropriate PSD estimates and to use them within the radar scatterometer equation to invert for σ^o values on the eight viewing angles and 3) to accept, store and transfer various data required by the processor or the experimenter.

Communication between the micro-processor and various peripheral systems is provided by the interface units indicated in Figure 5.1. Among these interface units is the operator interface and display. All system functions are initiated through this interface by the operator. The display is also used by the micro-processor.

The system of Figure 5.1 represents but a single architecture for the target system. It may not be the architecture of the final system but it will be close. Implied in Figure 5.1 is a single micro-processor to service both scatterometer channels. However, in view of the amounts of data and the number of computations involved, a single micro-processor may be unable to convert both channels of scatterometer to σ^o values at eight angles without large gaps between successive ground cells. A fast floating point processor such as the Advanced Micro Computer 95/4000 may permit use of a single processor. However, it may be necessary to dedicate a micro-processor to each scatterometer channel. To make an assessment of the number of processors requires that the software design be reduced to machine coding and that the machine be specified.

In view of this uncertainty and with full realization that a complete design effort requires an iterative effort between paper design (called for in this contract) and laboratory evaluation, an engineering model which is expandable to the target system was actually designed. The architecture of the proposed design is illustrated by the block diagram of Figure 5.2. The development model is restricted to single PSD estimation channel and a single micro-processor. The software is sufficiently general to permit processing of like or cross polarized data for any transmit polarization. However, the processor must be cued externally as to which polarization channel the PSD estimation subsystem is connected. A single channel processor of this type is sufficiently simple to fully evaluate the total system design. It can be readily expanded to a two channel system, terminating in either a single micro-processor or two micro-processors. The expansion simply requires that the controller signals be routed to two channels and that simple modifications be incorporated into the software whether one or two micro-processors be required.

5.2 An Overview of the Operation of the System

As indicated above, the scatterometer processing system consists of two major subsystems together with the necessary interfaces to permit communication with various peripheral devices. The relationships among the major subsystems and interface units was shown in Figure 5.2. An alignment generator, although not required during operation, was also shown.

When the system is powered, software control of the system is assumed by the micro-processor within the so-called RESET mode. Within

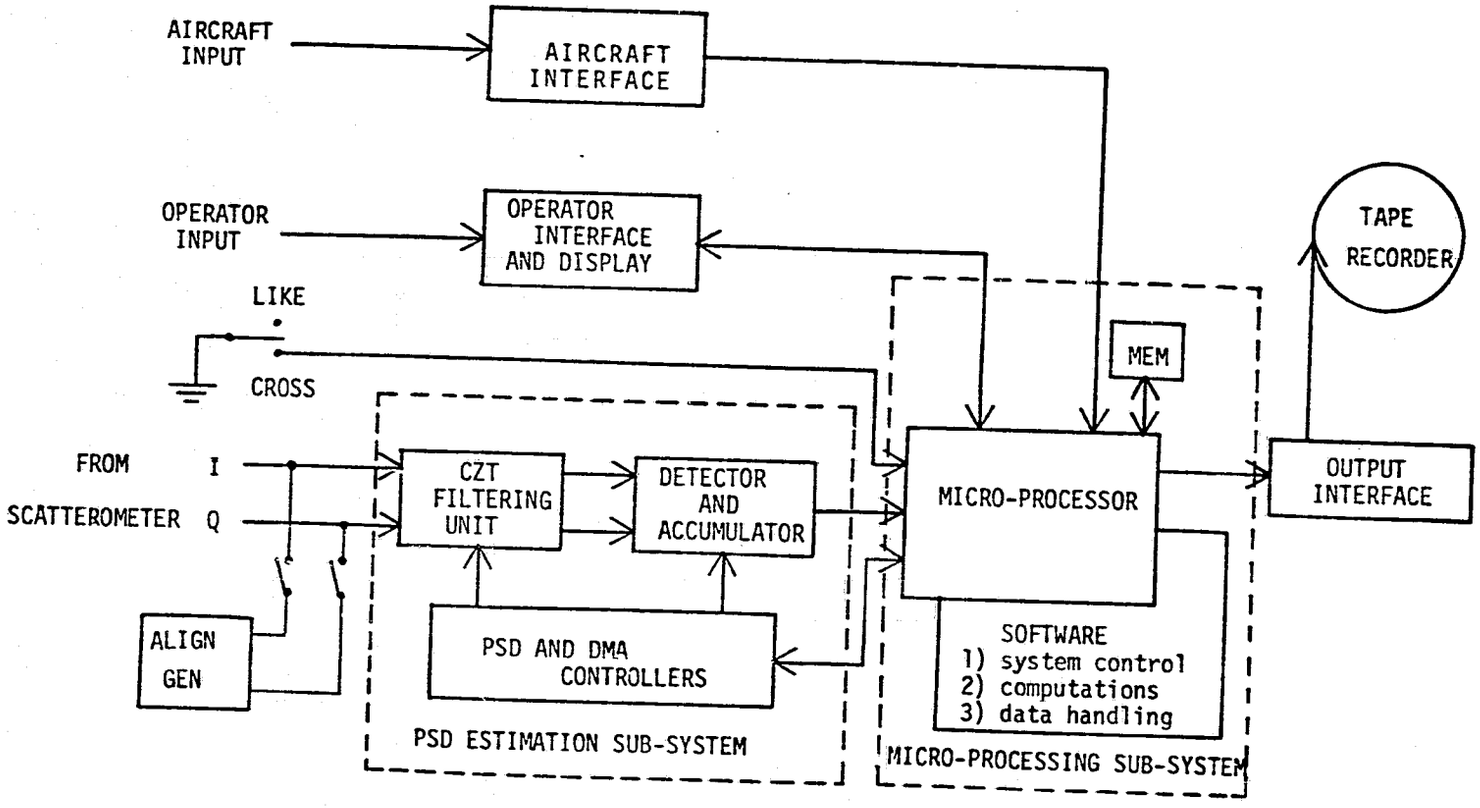


FIGURE 5.2 ARCHITECTURE OF AN ENGINEERING MODEL EXPANDABLE TO THE TARGET SYSTEM

this mode the aircraft, operator and output interface units, and the CZT filtering unit within the PSD estimation subsystem are active (see Figure 5.2). When the system is within this mode, the operator may override certain aircraft parameters arriving at the aircraft interface should they be inaccurate or missing in the aircraft data stream. From the RESET mode, scatterometer data may be processed on a flight line by entering the RUN mode.

Within the RUN mode, the software controller determines the number of 512 point sub-records to be processed to form an average return from the 512 beam resolution cells in view by the CZT filters. The number of sub-records N_R and the address to which the accumulated reserves are to be stored are transferred to the PSD controller (see Figure 5.2). Upon this initialization, the accumulator and detector are activated at the beginning of a CZT cycle. The PSD estimation subsystem then accumulates N_R estimates in each spectral channel. At the end of the N_R sub-records the detector and accumulator are halted and a DMA (direct memory access) transfer is made from the accumulator to the micro-processor memory. If the processor has not been halted by the operator, the detection and accumulation cycle is re-initiated by the micro-processor once the DMA has been completed. It should be noted that the filtering sub-section operates continuously between accumulations to avoid start up transients.

A closer look at the PSD estimation subsystem will indicate that this subsystem filters the fore and aft Doppler spectra of the radar return to form estimates of the return power is 512 parallel spectral channels. Of these channels, 256 are available to characterize the fore spectrum and 255 the aft spectrum. The remaining channel monitors

the return from the nadir point (this return is suppressed by the radar and the processor). A sliding CZT algorithm, as discussed in Section 3.5, is employed to form repeated estimates of the power return within each channel. Several estimates are summed to form an improved estimate of the average power.

The format of the summed spectral estimates in the 512 channels is illustrated in Figure 5.3. When the forward CZT is formed on the complex signal x + jy, the aft spectrum appears in the first 257 accumulation bins (channels) and the fore spectrum in reverse order in the latter 255 accumulation bins. Some of the spectral channels among the 512 channels are reserved for the calibration and polarization tones. The effective bandwidth of each channel is given by $B = f_s/512$ where f_s is the sampling frequency. The frequency resolution, i.e., the separation between spectral estimates is also equal to B. The normalized frequency response of each filter in dB is illustrated in Figure 5.4. The filter efficiency to the first sidelobes is 90.3% and through the first sidelobes is 95.5%*. The effective bandwidth is also equivalent to the width of the mainlobe.

The average spectral estimates together with calibration and polarization tone levels are transferred from the PSD estimation subsystem to the micro-processing subsystem through a DMA process initiated by the

---

*It is possible to increase the mainlobe efficiency by weighting the tap points on the CZT transversal filter. Such a device is available. However, this design is not recommended since 1) the precision will be reduced by a factor of two (the adjacent channel outputs are highly correlated) and 2) the S/N ratio referred to the output of the weighted transversal filter degrades by a factor of two (the noise is primarily governed by the post amplifier and signal output is reduced when weighting is used).

W = BANDWIDTH CORRESPONDING TO DESIRED ANGULAR RESOLUTION

B = FREQUENCY RESOLUTION AND EFFECTIVE BANDWIDTH

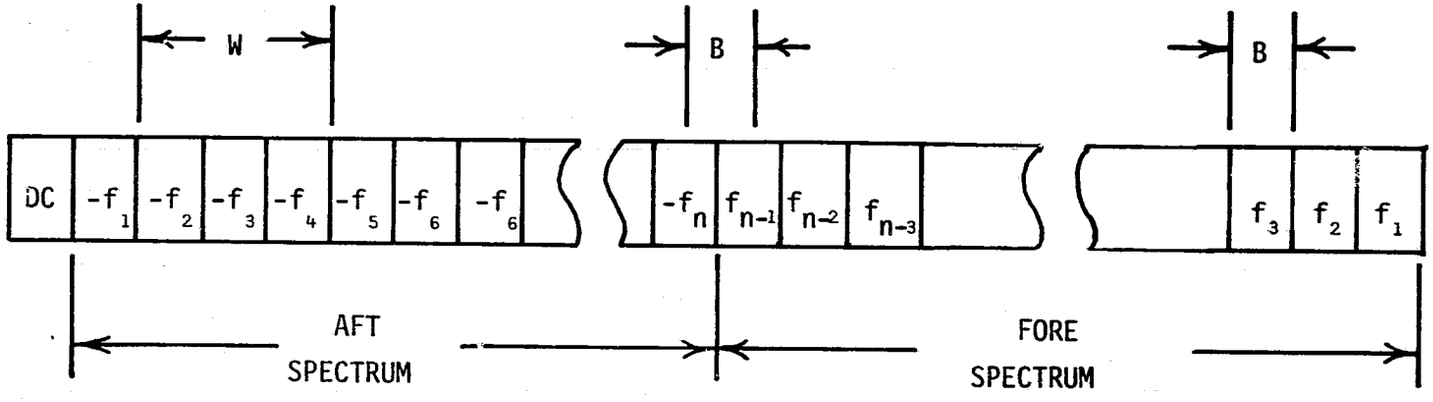


FIGURE 5.3 SPECTRAL DATA FORMAT AND RELATED PARAMETERS

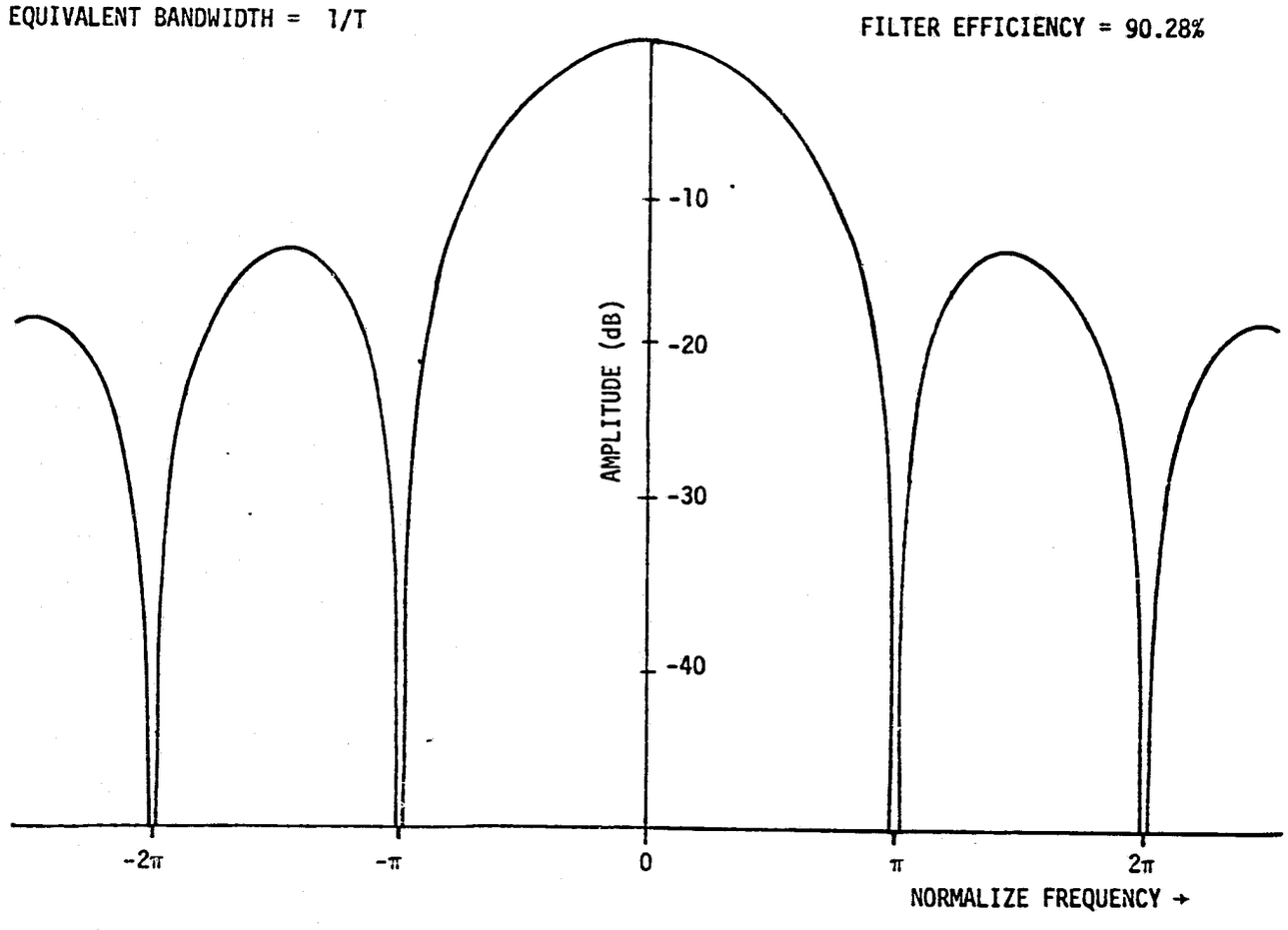


FIGURE 5.4 FILTER POWER SPECTRAL RESPONSE

PSD controller. Once the detector and accumulator are re-initiated, the micro-processor, using data from the aircraft interface together with invariant calibration constants stored in ROMs (Read Only Memories), converts the spectral and calibration data to σ^0 estimates at the required eight viewing angles. Adjacent spectral estimates are summed about each viewing angle to form a total return within the desired angular resolution. The spectral lines are chosen dynamically to track the specified viewing angle.

The processed scatterometer data and other parameters are stored in an array of the micro-processor memory for each of the viewing angles. The σ^0 estimates at the eight angles are stored in memory in an askewed fashion to provide near collocation of the returns on a single cell. When the re-ordered data is available on a single cell at all angles, it is written out to the output interface which in turn transfers it to magnetic tape. The transfer of the data continues until the operator halts the processor. When halted, the processor places an end of file indicator within the output array, transfers the partially filled array to tape and enters the RESET mode to await instructions from the operator. At the beginning of each accumulation cycle, appropriate parameters are withdrawn from the aircraft data channel for use in the computations and for transmittal to the output tape.

6.0 THE PSD ESTIMATION SUBSYSTEM

6.1 Introduction

As indicated in Figure 5.2 the PSD estimation subsystem consists of three sections:

1) the CZT (Doppler) filtering unit
2) the detector and accumulator unit
3) the PSD and DMA controllers.

Filtering in parallel channels is accomplished by performing a sliding chirp Z transform on a complex valued input signal in accord with the theory discussed in Sections 3.4 and 3.5. The technique for implementing the CZT is based on a CCD transversal filter and requires a configuration as described in Section 3.6. The filtering section employs analog and sampled analog techniques. The remaining sections are implemented digitally in order to achieve the necessary dynamic range and accuracy.

6.2 The CZT Filtering Unit

The CZT filtering unit is composed of four elements: 1) the input conditioning circuit, 2) the pre-chirp multipliers and summers, 3) the CCD transversal filter and 4) the signal extraction circuitry. Each element is discussed below. Detailed block diagrams for the four elements are found in Figures 6.1 and 6.3. A timing diagram for the associated clocking/gating signals is shown in Figure 6.4. The associated design schematics are included in a separate blue print file and part listings may be found in Appendix D.

6.2.1 Input Conditioning Circuit

The input conditioner consists of buffers, low pass active filters (LPF), sample and hold circuits, and a bank of variable gain amplifiers as illustrated in the block diagram of Figure 6.1.

The signal in each input channel is buffered through an ac coupled amplifier to totally remove dc and drift components induced by the scatterometer or recording system. It is important to remove these components since, when large, they can severely limit the dynamic range over which a PSD can be executed. This requirement is imposed by the input dynamic range to the CZT transversal filter which is somewhat less than $\pm$ 2.0 volts. The ac coupling causes a highpass filter break point at 88 Hz for the C band processor and at 28 Hz for the L band processor. Although the amplifier is ac coupled, an dc test signal may be injected into the buffer amplifier through a separate jack to align or test the CZT filtering unit.

After buffering the input signals in each channel, the upper bandwidth of the signal is established by a four pole maximally flat Butterworth filter. The design procedure for the filter is well known and may be found, for example, in Chapter 4 of reference [6]. The transfer function for the LPF in the C band system is given by

$$\frac{V_o(s)}{V_{in}(s)} = \left[\frac{1.59\ \omega_{oc}^2}{s^2 + \sqrt{2}\ \omega_{oc}\ s + \omega_{oc}^2} \right]^2$$

where $\omega_{oc} = 2\pi f_{oc}$ and $f_{oc} = 5.9$ KHz and by

$$\frac{V_o(s)}{V_{in}(s)} = \left[\frac{1.59\ \omega_{OL}^2}{s^2 + \sqrt{2}\ \omega_{OL} s + \omega_{OL}^2} \right]^2$$

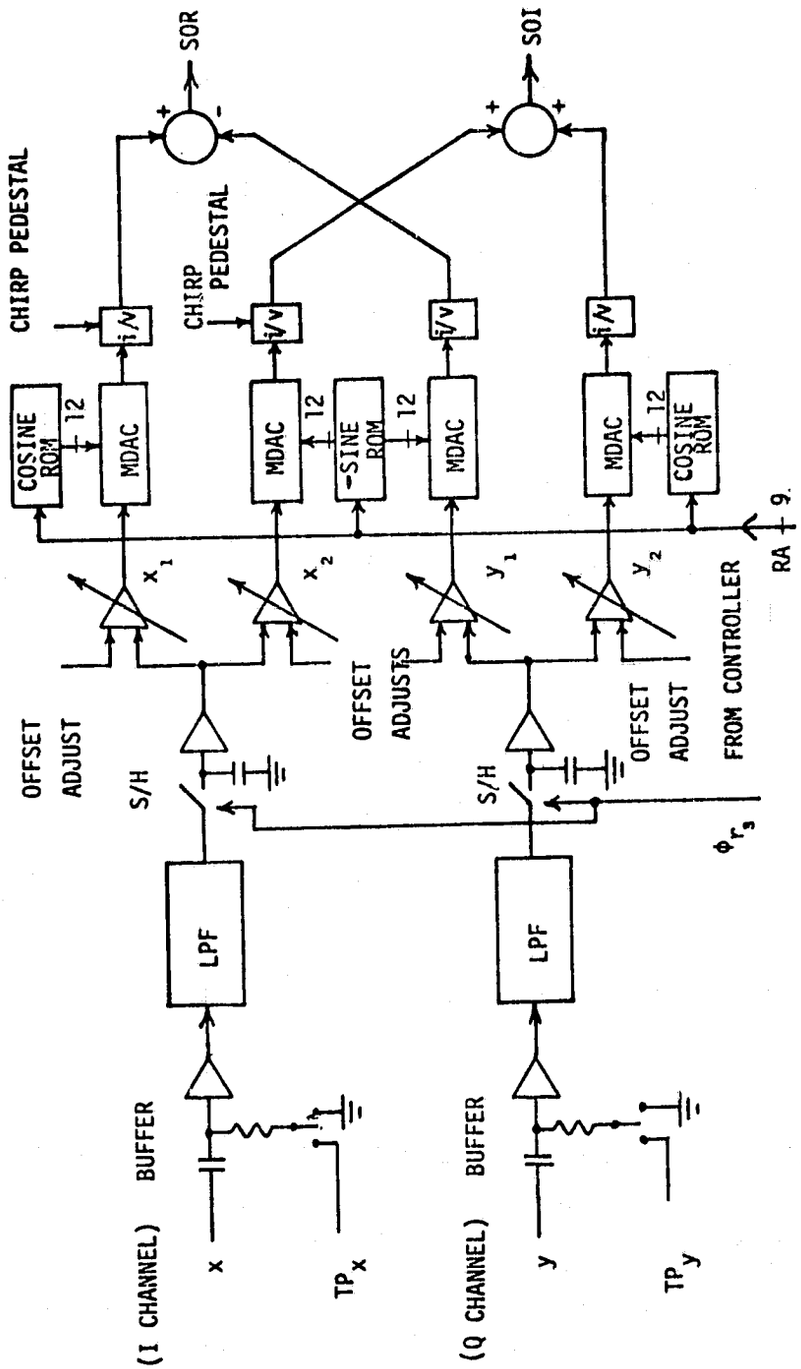


FIGURE 6.1 DETAILED BLOCK DIAGRAM OF CONDITIONING AND PRE-CHIRPING CIRCUITS

where f_{OL} = 4.0 KHz for the L band system. The low pass filters are required to establish the processing bandwidth prior to sampling to avoid aliasing of high frequency noise components into the desired spectrum.

An examination of the association design schematics will show that low drift and low offset operational amplifiers (LM308A) have been selected for the buffer and LPF stages. A low drift characteristic is assured in each stage by balancing the shunt resistances into the non-inverting and inverting terminals of the operational amplifier. This design approach is important to minimize the dc component before chirping.

With the bandpass established by the buffer and LPF, the signal is prepared for analog sampling. A sample and hold (S/H) circuit converts the filtered input into a staircase approximation. The signal is sampled at a frequency 3 times the highest calibration/polarization tone. Analog sampling is required to stablize the amplitude for the subsequent pre-chirp multiplication and entry into the CCD transversal filter.

An N channel MOSFET is employed as a series switch in the S/H circuit. Its substrate is biased at -8.2 VDC and its gate is protected with an external diode (see schematics). The sampling switches are actuated by ϕ_{R3}. A high impedance FET input operational amplifier is employed to buffer the sampling capacitor.

A pair of offset and gain adjustable amplifiers are employed to provide two parallel channels after each sampler. Two channels are required in each input channel to prepare the I and Q signals for complex valued multiplications and additions. The dc offsets in all four channels

should be suppressed to the millivolt level prior the pre-chirp operation. The gains must be adjusted to nominally provide a 10 volt signal into the chirp multipliers for the maximum recorder output (1.5 volts). Final gain adjustment should account for the gain differences through each multiplier, however. It is important to operate the channels at the highest permissible amplitude (accounting for signal fading) to produce multiplications with good S/N characteristics.

6.2.2 Pre-chirp Multiplication and Summing

As indicated in Section 3.4, the CZT requires that the complex valued input signal z be pre-chirped with $e^{-j\pi(k-n)^2/N}$ where (k-n) is the sample index*. Therefore the pre-chirp and summing operations must form

$$z_m e^{-j\pi m^2/N} = (x_m + jy_m) \quad (\cos\pi m^2/N - \sin\pi m^2/N) \tag{6.1}$$

where m plays the role k-n. When the real parts and imaginary parts are identified.

$$\text{Re}\ \{z_m e^{-j\pi m^2/N}\} = x_m \cos\pi m^2/N + y_m \sin\pi m^2/N \tag{6.2}$$

and

$$\text{Im}\ \{z_m e^{-j\pi m^2/N}\} = -x_m \sin\pi m^2/N + y_m \cos\pi m^2/N \tag{6.3}$$

The above results indicate that the I and Q channel signals must be multiplied by cosine and sine chirps and then algebraically summed.

---

*This may not be apparent to the casual reader. However, if the redundances of modulo 2π are removed, then the assertion stands. The sample index here is unlimited since a sliding transform is employed.

The stratagem by which this is accomplished is illustrated in the block diagram of Figure 6.1.

Multiplying digital to analog converters (MDACS) are employed to perform the pre-multiplication operation. The MDACS, in conjunction with the current to voltage converters and the read only memories (ROMS), are required to form the products cited in equation 6.1. An MDAC is composed of a resistor ladder network whose shunt currents may be diverted in either of two output ports. When these ports operate into virtual grounds as provided by an operational amplifier circuit, the current at output port #1 is given by

$$I_1 = I_{in} (a_1 2^{-1} + a_2 2^{-2} + \ldots + a_n 2^{-n})$$

and at the second port by

$$I_2 = I_{in} - I_1$$

where $a_k \varepsilon \{0,1\}$ and $k \varepsilon \{n\}$ and $I_{in} = 10^{-4} V_{in}$. The a_k's are externally selectable by appropriately addressing the ROM and storing the binary number $\{a_1, a_2, \ldots a_n\}$ at that memory location. When these currents are sunk into the virtual grounds of the operational amplifier network shown in Figure 6.2, the output voltage is given by

$$V_o = I_{in} R_2 (1-2K)$$

where $K = a_1 2^{-1} + a_2 2^{-2} + \ldots + a_n 2^{-n}$. To produce a cosine chirp (i.e., the I_{in} must be multiplied by a cosine chirp), taps on the ladder network must be selected so that

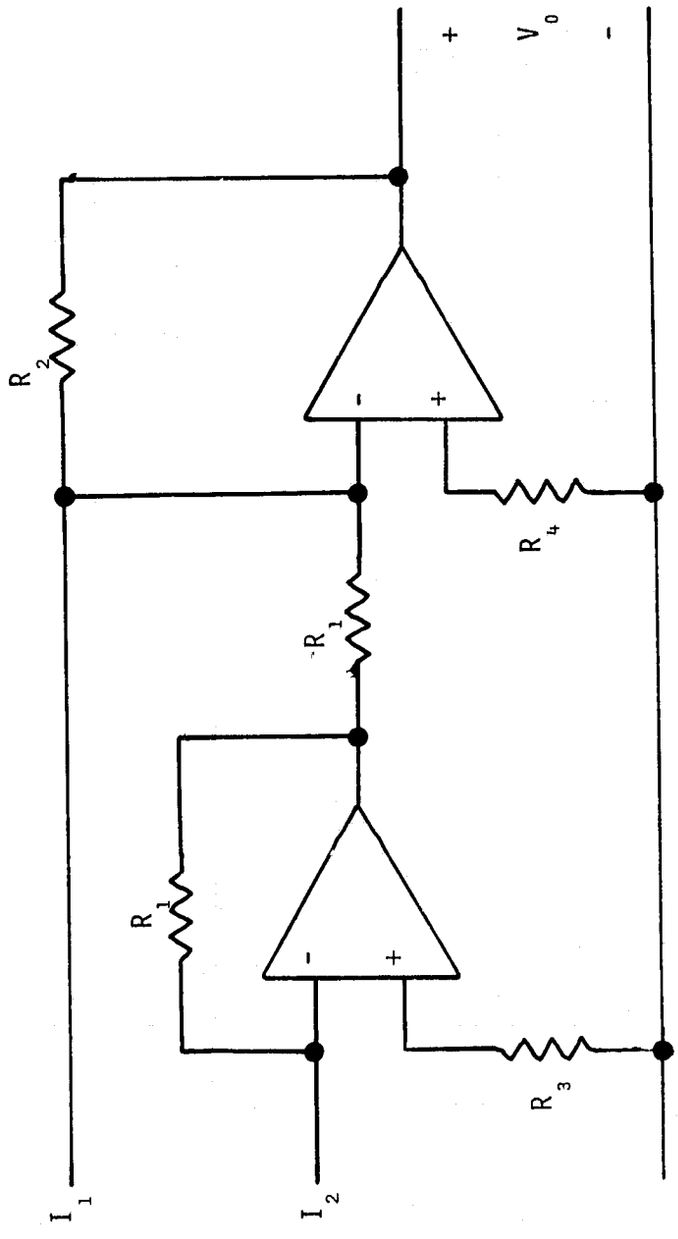


FIGURE 6.2 CURRENT TO VOLTAGE CONVERTER

$$1-2K_c = \cos\pi m^2/N$$

or

$$K_c = \frac{1-\cos\pi m^2/N}{2}$$

when the mth sample is inserted. Similarly to produce a -sine chirp, the MDACS must be addressed so that

$$K_s = \frac{1 + \sin\pi m^2/N}{2}$$

on the mth sample. Since $0 \leq K \leq 1$, it is clear that the chirps can be achieved. The proper values of K_c and K_s are stored sequentially in a ROM and withdrawn synchronously with the input samples. A FORTRAN program which generates the closest binary values of K_c and K_s throughout a single chirp cycle is shown in Table 6.1. The values of K_c and K_s are printed in hexidecimal and their BCD equivalent.

As a note of caution, it is important that the operational amplifiers employed in the current to voltage converters have sufficient bandwidth and slew rate to support sudden reversals in polarity. The LF 356 operational amplifier specified in the design schematics meet the bandwidth and slew rate requirements. Their low noise performance also makes them attractive in these circuits since the signals are limited by the current capability of the MDACS.

After the I and Q signals are appropriately chirped, differencing and summing amplifiers are employed to form the real and imaginary parts required by equations 6.2, and 6.3, respectively. Prior to the summing and

```
C
C
C                    PROJECT 2556
C                    REMOTE SENSING CENTER
C                    TEXAS A&M UNIVERSITY
C
C
C       THIS PROGRAM GENERATES OFFSET BINARY FOR PRE-CHIRP FUNCTIONS
C       THIS VERSION GENERATES THE DOWN CHIRP
C
C                         MAIN PROGRAM
C
      1       REAL MSCOS,MSSIN
      2       DATA PI/3.141593/
      3       AVECOS = 0.0
      4       MSCOS = 0.0
      5       AVESIN = 0.0
      6       MSSIN = 0.0
      7       WRITE(6,105)
      8   105 FORMAT('1',T55,'REQUIRED VALUES',T83,'BCD EQUIVALENT',T118,'ERROR
     *  ',/)
      9       WRITE(6,100)
     10   100 FORMAT(' ',T10,'ADDRESS',T20,'SINSIN',T30,'SINCOS',T40,'COSCOS',
     *  T55,'XSIN',T65,'XCOS',T81,'ISIN',T95,'ICOS',T110,
     *  'ERSIN',T125,'ERCOS'//)
     11       DO 10 J=1,512
     12       I=J-1
     13       X = FLOAT(I)
     14       XCOS = COS(PI*X*X/512.0)
     15       XSIN = -SIN(PI*X*X/512.0)
     16       ICOS =  2048 - IFIX( 2048.0*XCOS + SIGN(0.5,XCOS))
     17       IF ( ICOS .GT. 4095)  ICOS = 4095
     18       ISIN =  2048 - IFIX( 2048.0*XSIN + SIGN(0.5,XSIN))
     19       IF ( ISIN .GT. 4095)  ISIN = 4095
     20       ERCOS = (2048.0 - FLOAT(ICOS))/2048.0 -XCOS
     21       ERSIN = (2048.0 - FLOAT(ISIN))/2048.0 -XSIN
     22       AVECOS = AVECOS  + ERCOS/512.0
     23       AVESIN = AVESIN + ERSIN/512.0
     24       MSCOS = MSCOS + ERCOS*ERCOS/512.0
     25       MSSIN = MSSIN + ERSIN*ERSIN/512.0
     26       ICC=MOD(ICOS,256)
     27       ISC=MOD(ISIN,16)*16+ICOS/256
     28       ISS=ISIN/16
     29    10 WRITE(6,200) I,ISS,ISC,ICC,XSIN,XCOS,ISIN,ICOS,ERSIN,ERCOS
     30   200 FORMAT(T12,Z3,T19,'|',T22,Z2,T32,Z2,T42,Z2,T49,'|',T52,F8.5,T62,F
     *  8.5,T74,'|',T77,I6,T91,I6,T103,'|',T107,E10.3,T122,E10.3)
     31       MSSIN = SQRT( MSSIN - AVESIN*AVESIN)
     32       MSCOS = SQRT( MSCOS - AVECOS*AVECOS)
     33       WRITE ( 6,300 )  AVESIN,MSSIN
     34   300 FORMAT(///T12,'AVERAGE SINE ERROR = ',E15.5,5X,'MEAN SINE ERROR =
     *    ',E15.5/)
     35       WRITE ( 6,400 )  AVECOS,MSCOS
```

TABLE 6.1 CHIRP GENERATION PROGRAM

differencing operations, dc offsets are introduced into the real and imaginary channels since the CCD transversal filter requires that the chirped signals be offset on a positive voltage.

6.2.3 The Transversal Filter and Signal Extraction Circuitry

The real and imaginary parts of the chirped signals are injected into the imaginary and real channels of the CCD transversal filter, respectively*. Two transversal filters are required on each input to perform the complex valued convolution required by the CZT.

With the connections as illustrated in Figure 6.3, the quad filter bank is matched to a down chirp. The reversal in sign in one of the sine channels is purposely included in the bank so that both of the output channels can be differentially connected. A differential connection is required because the convolution outputs must be extracted by pulsing the ϕ_1^+ and ϕ_1^- and the ϕ_3^+ and ϕ_3^- outputs through coupling capacitors C_o (see Figure 6.3). Once a differential output is pulsed, the capacitors act as integrating capacitors to receive the output charges from the transversal filters. Prior to receiving each new convolved output, the previous output sample is discharged using MOSFET switches within the CCD chip. The switches are activated by the ϕ_{R1} and ϕ_{R3} gating signals. Once the integrating capacitors have been discharged, the output on the $\phi_1^{+}_{-}$ channel is extracted with gating signal ϕ_1 and the output on the $\phi_3^{+}_{-}$ channel is extracted with gating signal ϕ_3. The combination of signals

*The CCD filter is masked to perform the inverse DFT. This requires that the inputs be transposed to produce the forward transform as required by this design.

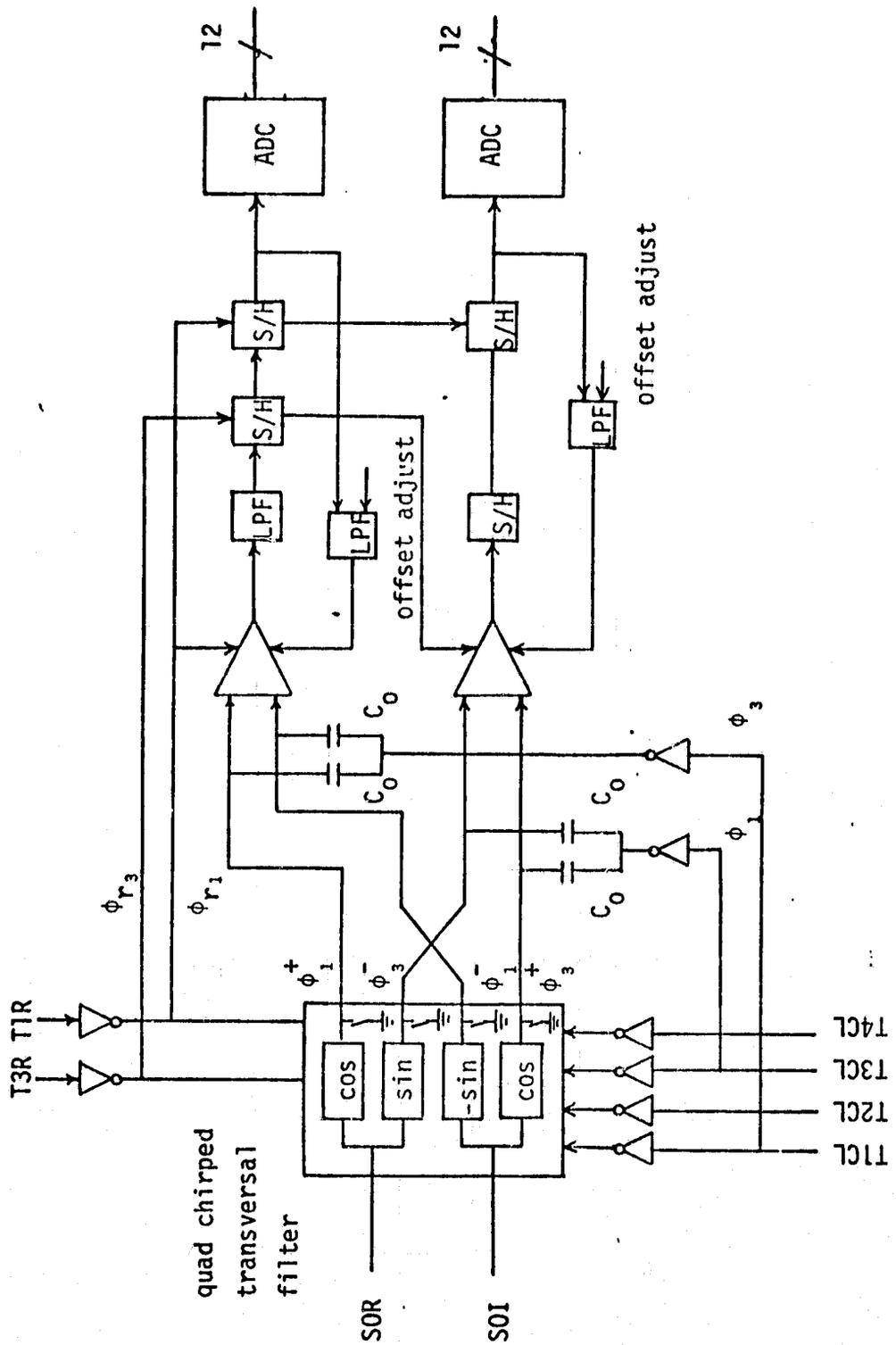


FIGURE 6.3 DETAILED BLOCK DIAGRAM OF TRANSVERSAL FILTER AND SIGNAL EXTRACTION CIRCUITRY

ϕ_1, ϕ_2, ϕ_3, and ϕ_4 are required to move the input signals down the transversal filters. The timing diagram for these gating signals is shown in Figure 6.4.

An overview of the sampling and convolution operations can be established from the timing diagram. A basic sample period is broken into 64 sub-periods and the different segments of sub-periods are allocated to the timing signals. A new input sample is taken when ϕ_{R3} is high. This same signal discharges the previous ϕ_3 output signals on C_o. This input sample is pre-multiplied by the next chirp value clocked from the ROMS by CZTCLK. This pre-multiplied sample is loaded into the transversal filter with ϕ_2. Simultaneously ϕ_2 also latches the previous convolved output on the ϕ_3 channel into an analog output buffer. This output is extracted when ϕ_3 goes high. At the end of the ϕ_3 signal, the previous ϕ_1 sample on the ϕ_1^+ channels is discharged when ϕ_{R1} goes high. Then ϕ_4 latches the newly convolved output on the ϕ_1 channel in its output buffer. The ϕ_1 output is extracted when ϕ_1 goes high. The cycle then repeats.

It is important to note that the transversal filter is actually a filter matched to a down chirp. Since all frequency components are down chirped in the pre-chirp operation, each frequency component is compressed by a factor 512 in the matched filter. This implies that frequency components not "visible" on the input when chirped and compressed will be visible at the output of the transversal filter. As a result of this processing gain (27dB), the dynamic range of the transversal filter is quite impressive and is primarily limited by the noise of the post amplifier [7].

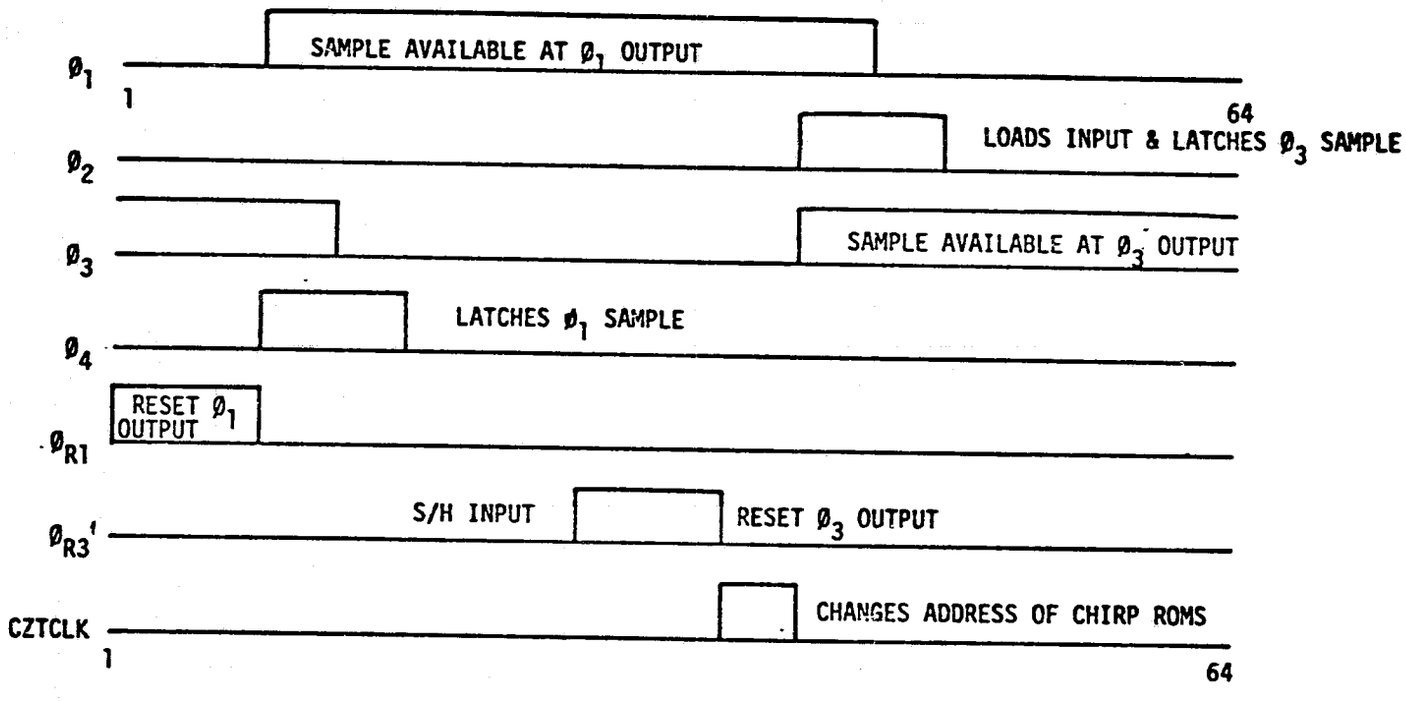


FIGURE 6.4 TIMING DIAGRAM FOR CZT FILTERING UNIT

The two output channels except for a constant phase factor contain the real and imaginary estimates of the DFT of the complex valued input signal. The output voltage is held and buffered by a sampling differential amplifier. The integrating capacitors must be well balanced to retain the common mode rejection inherent in the differential amplifier. This assures that the extraction signal is rejected from the output of the amplifier.

It is important to employ low noise operational amplifiers at the outputs of the transversal filter to realize the inherent dynamic range of the filter. Since low noise amplifiers typically have large bandwidths associated with them, the differential stages are followed by low pass amplifiers to reduce the noise bandwidth without affecting the signal bandwidth. The low pass amplifiers are followed by a series of S/H circuits. The ϕ_1 channel has an additional S/H to bring that channel into synchronism with the ϕ_3 channel. The S/H stages also function to remove the residual common mode signal. This signal would appear at the output of the S/H chain as a dc offset if no dc compensation were employed. However, to remain calibrated at the output a dc stabilization loop has been included to maintain the <u>average</u> output near zero volts. A null adjustment is available in each output channel to achieve this condition. The negative feedback is very effective in removing and stabilizing the dc component before digital conversion and allows the output channels to operate over a full dynamic range of $\pm$ 10 volts.

The output of each S/H chain is digitized to 12 bits by an analog to digital converter (ADC). The 12 bits allow a dynamic range of 41 dB with a resolution of $\pm$ 0.5 dB at the lower end of the dynamic range based on a single record.* The output of the ADC appears as complementary offset binary but is converted to 2's complement by inverting the lower 11 bits (see Figure 6.5).

6.3 The Detection and Accumulation Unit

The digitized output DO3 and DO1, except for a post chirp phase factor, represent the real and imaginary parts of the DFT of the complex valued input signal. To convert these to a PSD, the real and imaginary parts must be squared and repeated estimates of the same spectral line averaged. These operations are equivalent to square law detection and accumulation of each spectral line. Within the PSD estimation subsystem, detection and accumulation are accomplished digitally in a manner illustrated by the block diagram of Figure 6.5.

The real and imaginary parts are squared with 12 bit multipliers to yield a 23 bit result. The squares are summed and passed into an accumulation loop. At the beginning of an accumulation cycle, previously accumulated results are withdrawn from memory, ANDed with zeros, and added to new spectral estimates. The ANDing operation results in adding zeros to the first of spectral estimates and therefore initializes the accumulation cycle. The results of the addition are latched in a 32 bit write register, routed through a read/write switch and stored sequentially in

*Note the total dynamic range is 66 dB; however, at low signal levels the experimenter cannot resolve $\pm$ 0.5 dB over this range. However, some of this larger range is required for signal fluctuations.

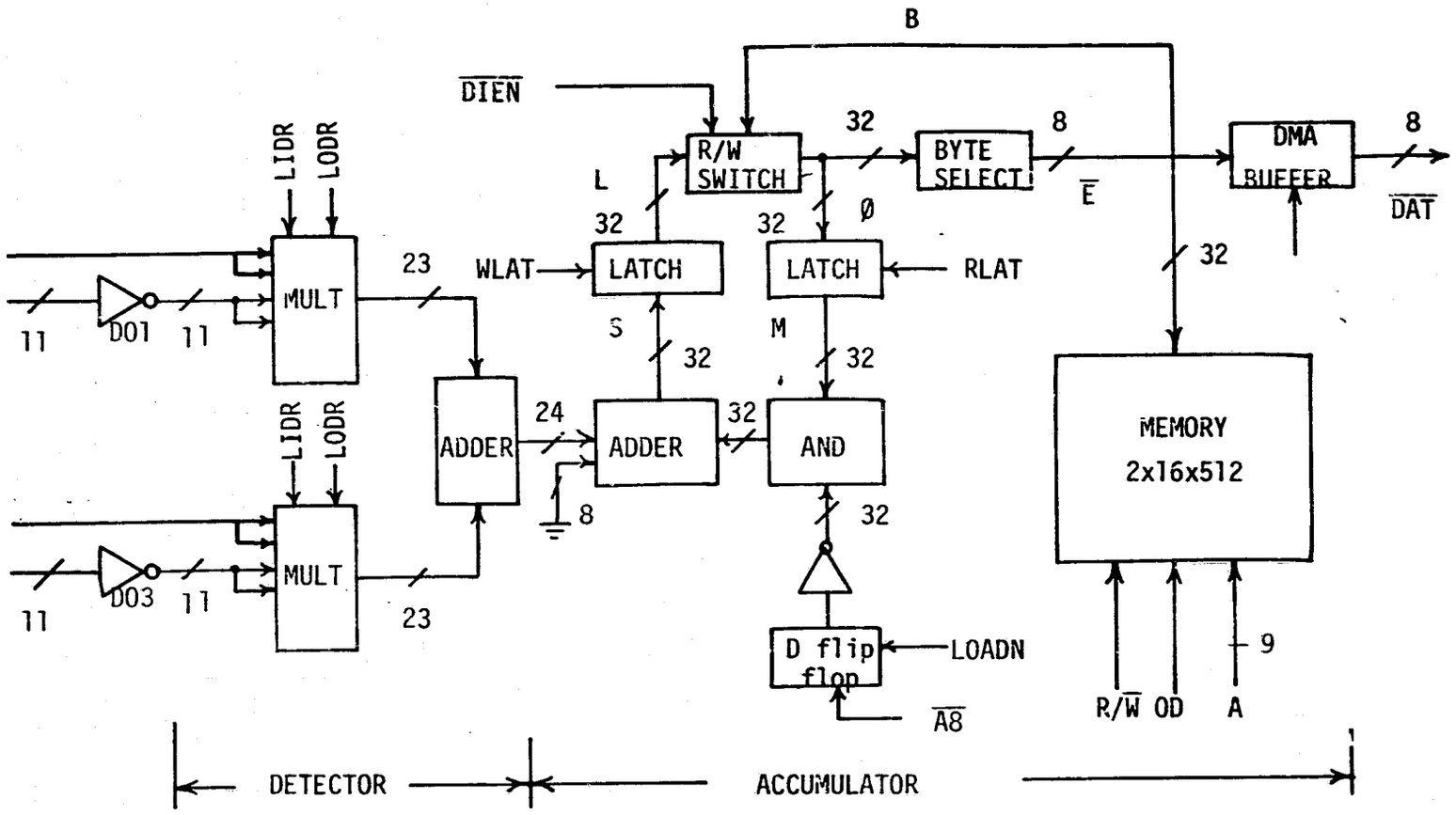


FIGURE 6.5 DETAILED BLOCK DIAGRAM OF THE DETECTION AND ACCUMULATION CIRCUITRY

memory. The next sample to be updated is then withdrawn from memory, directed through the read/write switch into the read latch. The output of the latch is ANDed with zeros until the initialization cycle has been completed.

When additional sub-records are to be accumulated beyond the initial sub-record, the sequence of operations remains the same; however, the previous spectral estimate is ANDed with ones instead of zeros and then added to the new spectral estimate. This operation continues until the desired number of sub-records have been accumulated. The accumulation loop will permit accumulations up to 32 bits. For a homogeneous target this corresponds to averaging up to 256 sub-records.

When the accumulation over the entire record has been completed, the contents of the 2x16x512 memory are unloaded sequentially under request by the DMA controller. The words from memory are transferred through the read/write switch to the read channel. Byte selection signals ϕA0 and ϕA1 decompose the word into sequential bytes, which are then latched into the DMA buffer whenever the data bus is not busy.

The sequence of operations for the detection/accumulation processor may be best understood from the timing diagram of Figure 6.6. During the 64 state cylce, the next spectral estimate is sampled when $\overline{S}$/H goes low. An A to D conversion is initiated on the negative transition of the START CONVERT command. The digital sample is loaded into input register of a multiplier by the Command LDIR. The product is loaded into the output register of the multiplier by LDOR. $\overline{DIEN}$ directs the contents of memory location A into the read loop. The word is latched by RLAT so as

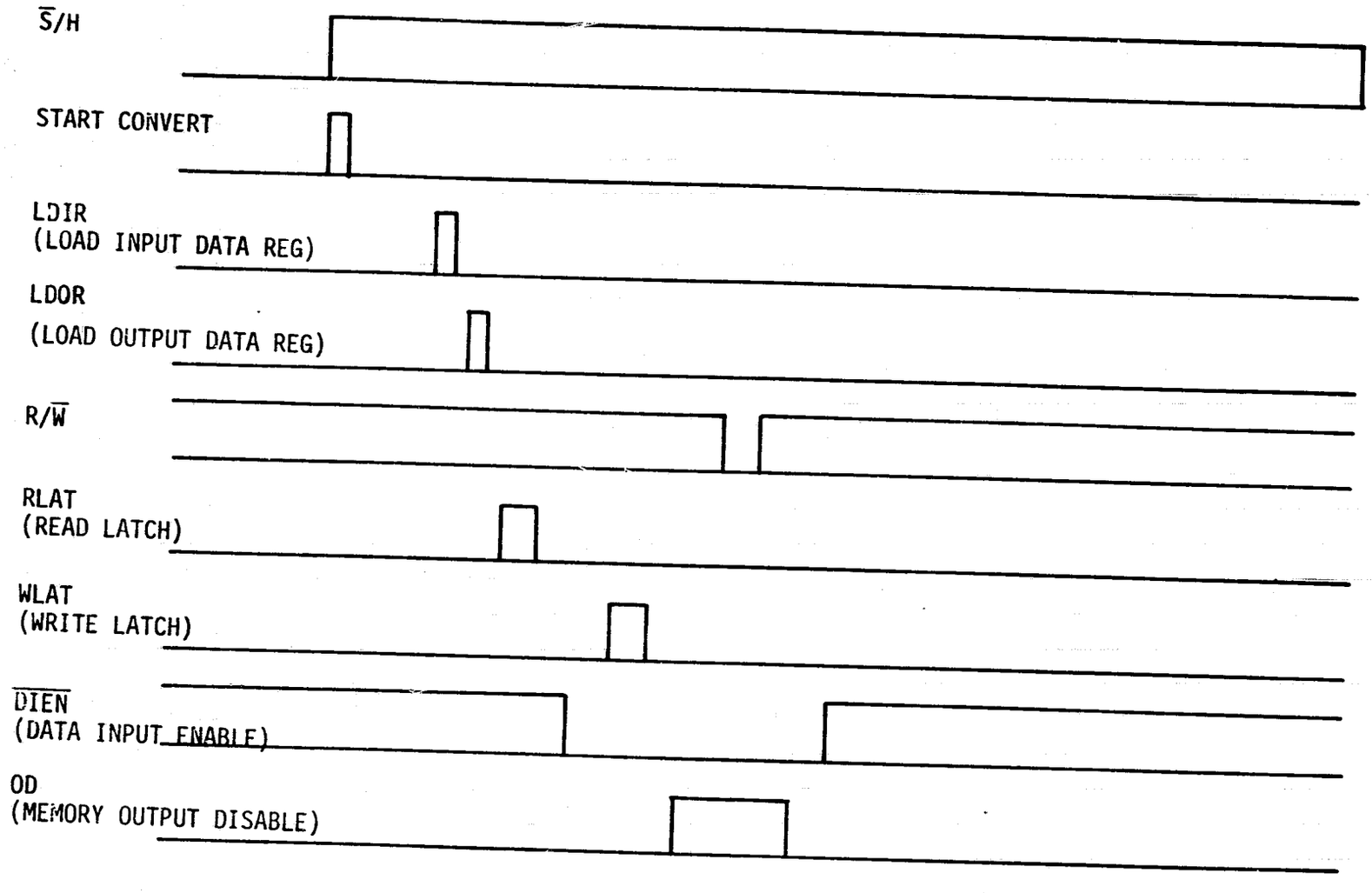


FIGURE 6.6 THE TIMING DIAGRAM FOR THE DETECTION AND ACCUMULATION UNIT

to permit ANDing with zeros or ones. The accumulation is performed between RLAT and WLAT and the result is latched by WLAT. The read/write switch is directed to write when $\overline{DIEN}$ goes low. The memory output function is disabled by OD and a write occurs when R/$\overline{W}$ goes low.

6.4 The Processor Controller Subsystem

The processor controller subsystem is shown in Figure 6.7. The subsystem can be further divided into a PSD controller and a DMA controller. The PSD controller provides all of the timing and control signals for the analog chirp Z-transform circuit and the digital accumulation circuit. The DMA controller provides for rapid transfer of the PSD data from the digital accumulator memory to the main processor memory. Detailed descriptions of the operation of the PSD and DMA controller are presented in the following sections.

6.4.1 The PSD Controller

The PSD controller network performs two interrelated functions. The network generates a set of timing and control waveforms for the analog CZT board and a set of timing and control signals for the digital accumulation board. The PSD controller is implemented as a 64 state subsystem that is driven by the signal CCLK, the microprocessor system clock.

The CCLK signal is divided by 16 to provide a signal called MCLK. The frequency of MCLK is 576 KHz which provides a minimum state time interval 1.74 μs. MCLK increments a 6 bit counter whose outputs provide addre: ng for the 64 state controller read only memory (ROM). There are 16 state control) signals generated by the controller ROM.

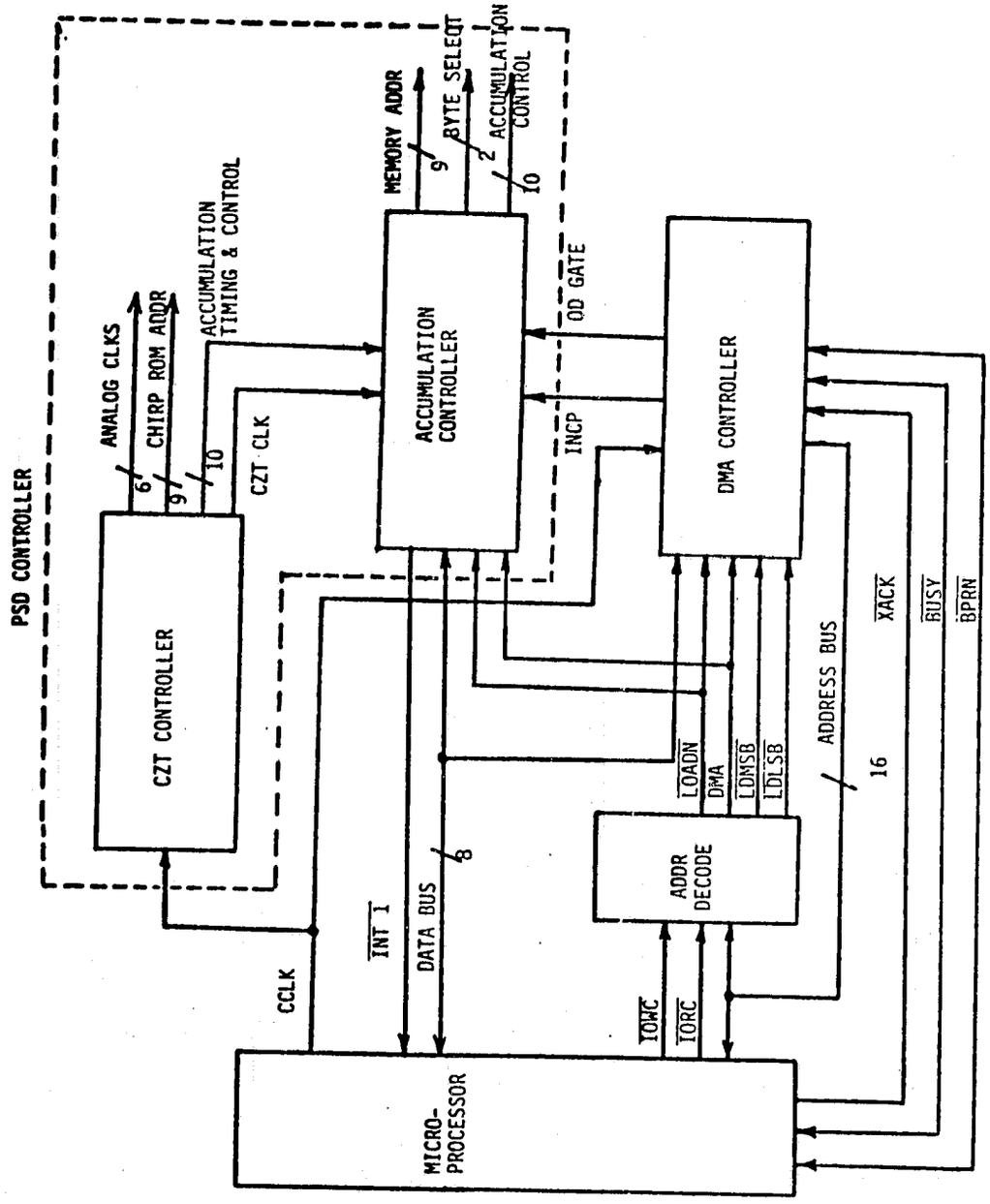


FIGURE 6.7 PROCESSOR CONTROLLER SUBSYSTEM

The state control signals are buffered by a data latch which is driven by $\overline{MCLK}$. Buffering is provided to eliminate transition noise from the ROM outputs and to assure that all signal transitions are in exact time coincidence. Of the 16 state control signals, 6 are clock signals which operate the CCD transversal filter, 9 are required for A/D conversion, squaring, and digital accumulation, and one, CZT CLK, is utilized by both analog and digital circuits.

The 6 analog control signals are identified in Figure 6.8 as TICL-T4CL, TIR and T3R. These signals are supplied to the CZT board, which in turn, generates the signals ϕ_1 - ϕ_4, ϕ_{R1} and ϕ_{R3} as described in section 6.2. The signal CZT CLK increments a 9 bit chirp ROM address counter. The resulting address signals, RA0 - RA8, select the appropriate chirp waveform values to be multiplied with the sampled input signals as the first step in performing the discrete Fourier transform.

The digital control signals are also shown in Figure 6.8. The timing relationship for the signals are depicted in Figure 6.6. With the exception of $S/\overline{H}$, the digital control signals are shown to be gated off during the DMA transfer operation.

There are 9 address lines required to address the digital accumulation network memory. These address lines are also generated as the output lines of a 9 bit counter as depicted in Figure 6.8. The accumulator address memory counter is increment by one of two possible clocking signals. During data sampling and accumulation the memory address counters are clocked in synchronism with the CZT filter by the gated CZT CLK signal. During the data transfer operation the address counter is clocked at a much more rapid rate as determined by the signal INCP.

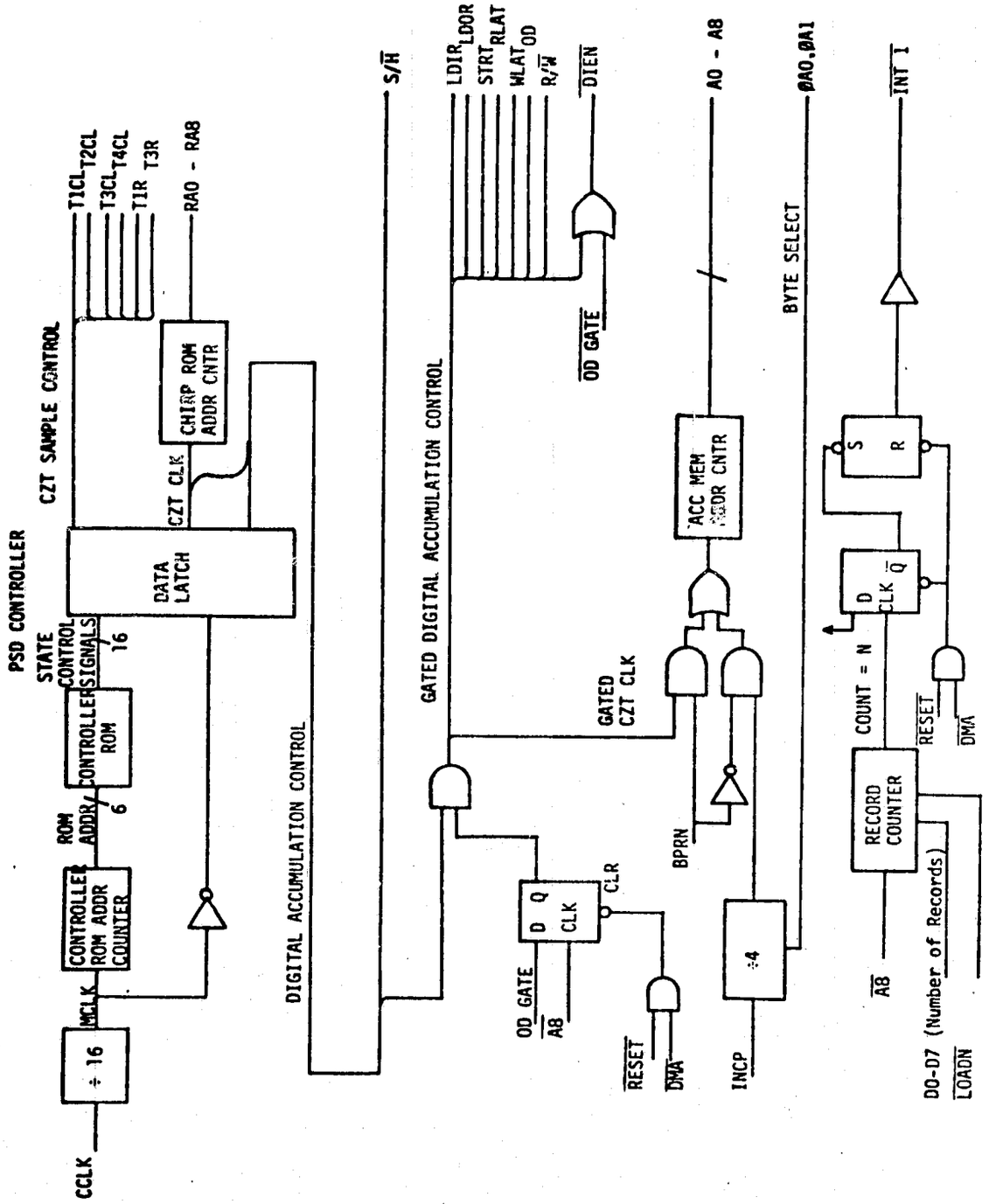


FIGURE 6.8 THE PSD CONTROLLER

During DMA only a single 8 bit byte can be transferred at a time so that each 32 bit data word must be transferred as four 8 bit data bytes. To identify the appropriate byte within each data word, two additional address, or byte select, lines are required. The byte select lines change with each cycle of INCP, whereas INCP must be divided by four to provide the clock signal drive for the memory address counter.

The transfer of the control of the accumulator memory and address counter is implemented by means of a record counter followed by two flip flops. These circuits generate an interrupt signal, $\overline{\text{INT 1}}$, that signals the end of a data accumulation cycle and that the DMA operation may begin.

To begin an accumulation cycle, an OUT instruction from the microprocess preloads a counter with the number of records that are to be added together or averaged. Upon the LOADN command the accumulation process begins. Each rising edge of the most significant bit of the address memory counter, $\overline{\text{A8}}$, decrements the record counter. When the count equals zero, the next rising edge of $\overline{\text{A8}}$ causes the counter to generate a "borrow" signal. The borrow signal in turn sets the DMA request flip flop which the processor observes as an interrupt. The processor services the interrupt by means of an OUT instruction to address 09. The signal generated by the OUT instruction is called $\overline{\text{DMA}}$. This signal activates the DMA operation.

6.4.2 The DMA Controller

The functions of the DMA Controller are to acquire control of the microprocessor multibus, transfer the accumulated PSD data to the processor main memory and return control of the multibus to the microprocessor.

Gaining and relinquishing control of the microprocessor multibus is accomplished by the circuit shown in Figure 6.9. The processor responds to a DMA request interrupt with the signal $\overline{DMA}$. This action causes the "bus priority in" signal $\overline{BPRN}$ to go high. $\overline{BPRN}$ prevents the microprocessor CPU from accessing the multibus during the data transfer operation.

In order for the DMA controller to transfer the data, the CPU must first relinquish control of the bus. This event is signaled to the DMA controller by allowing the open collector signal $\overline{BUSY}$ to go high. The DMA controller seizes control of the bus with the rising edge of $\overline{BUSY}$, and indicates its control by asserting $\overline{BUSY}$ back low. At the conclusion of the DMA transfer, the signal labeled DMA COMPLETE initiates action by the controller to return control of the multibus to the CPU.

In addition to initiating bus control transfer, the signal $\overline{DMA}$ inhibits the accumulation control signals to the digital accumulator board and forces the signal, OD GATE, low. OD GATE controls the digital accumulator memory data bus multiplexor such that data may be read from the accumulator memory and it also provides a means for synchronizing the accumulation cycle with the start of the chirp filter cycle whenever a new block of data is to be taken.

Once control of the multibus is established, the circuits in Figures 6.10 and 6.11 provide for the actual data transfer. The circuit in Figure 6.10 is driven by the computer clock signal $\overline{CCLK}$. During a DMA transfer cycle $\overline{BPRN}$ is high and the Q output of the transfer inhibit flip flop is also high. Under these conditions the clock signal, $\overline{CCLK}$, propagates through the three input AND gate, is divided by 8, and subsequently clocks

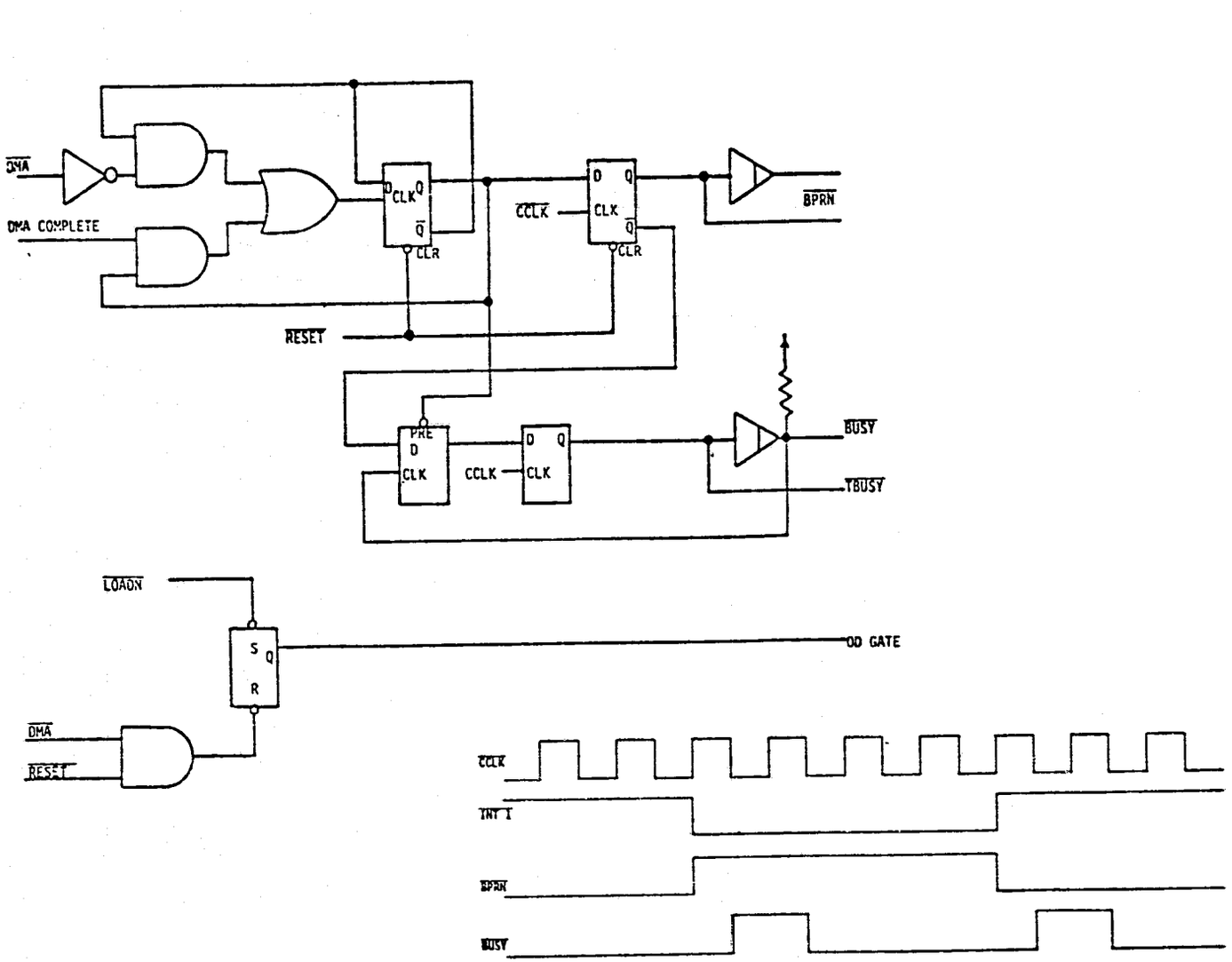
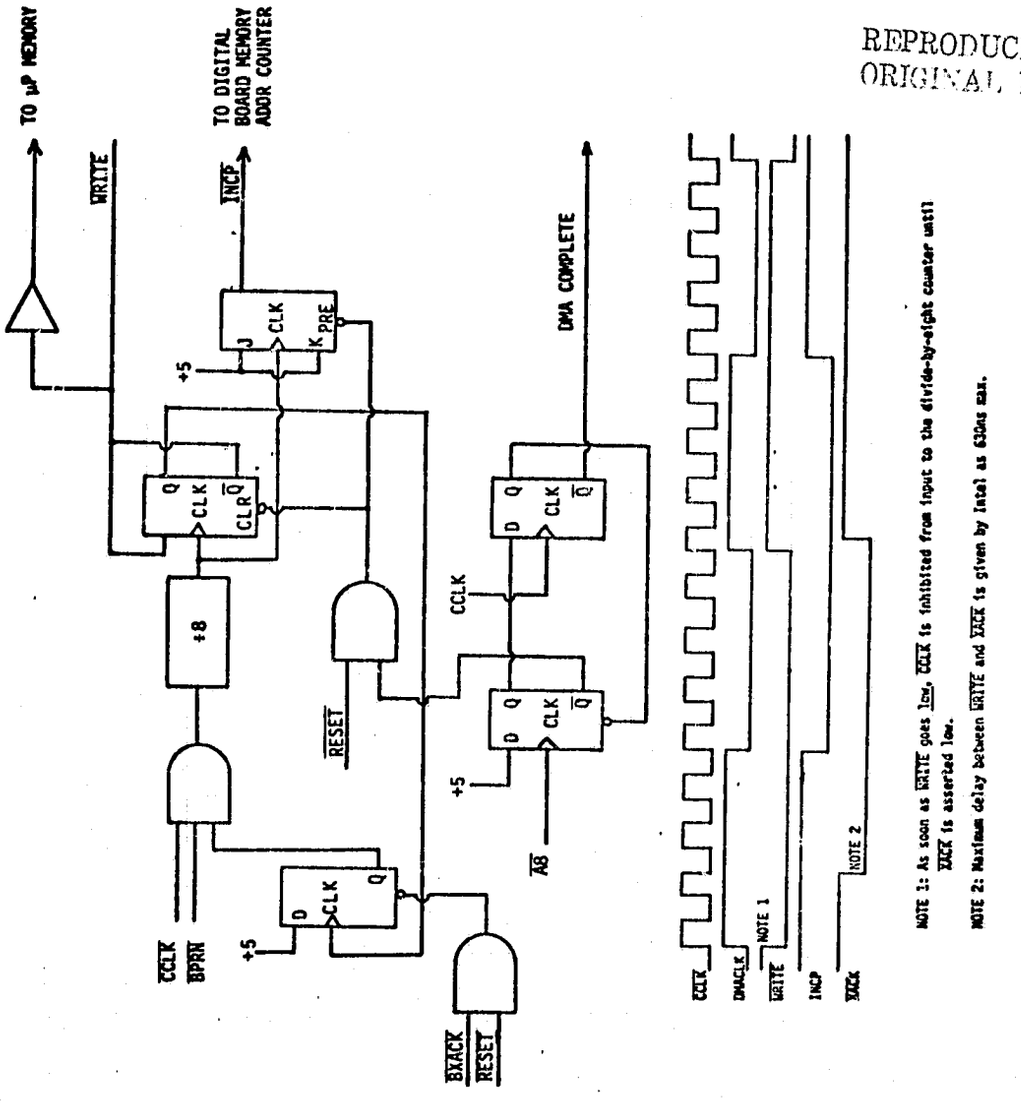


FIGURE 6.9 DMA BUSS PRIORTY CONTROLLER



NOTE 1: As soon as WRITE goes low, CCLK is inhibited from input to the divide-by-eight counter until XACK is asserted low.

NOTE 2: Maximum delay between WRITE and XACK is given by Intel as 630ns max.

FIGURE 6.10 DMA TRANSFER CONTROL

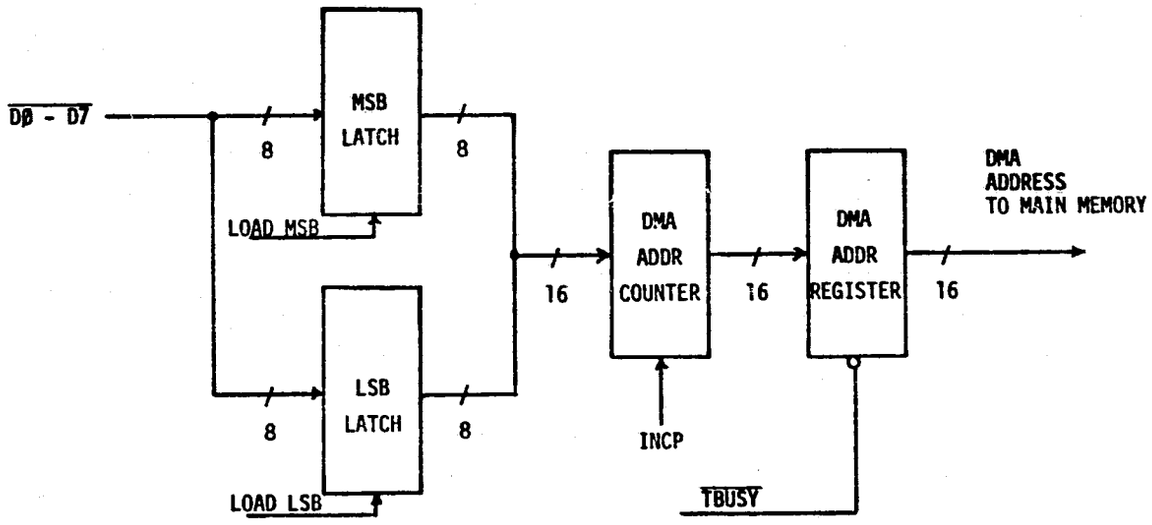


FIGURE 6.11 DMA CONTROLLER ADDRESS GENERATION

a D and J-K flip flop, both connected in a toggle mode. The D flip flop toggles on a rising edge of the divided clock signal whereas the J-K flip flop toggles on the falling edge.

The divide by 8 circuit first generates a falling edge clock signal so that the J-K flip flop toggles before the D flip flop. The Q output of the J-K flip flop is designated as the signal INCP. INCP increments the byte select segment of the accumulator memory address register, and on every fourth transition, increments the memory address.

After the proper memory address is provided, the D flip flop is toggled and a $\overline{\text{WRITE}}$ control signal is generated which commands the processor memory to store the PSD data present on the data bus at the memory location specified in the DMA controller address register. The complement of $\overline{\text{WRITE}}$ clocks the transfer inhibit flip flop to prevent the $\overline{\text{CCLK}}$ signal from incrementing to the next accumulator memory location until the write operation has been acknowledged by the microprocessor main memory controller.

The memory controller acknowledges the data transfer via the signal $\overline{\text{BXACK}}$. Upon receiving the transfer acknowledge the controller is again activated and the process continues until all 512 PSD samples are transferred. When the last sample is transferred, the signal DMA COMPLETE is generated and the DMA controller releases control of the multibus and is ready to begin a new DMA transfer cycle when a new record of data is received.

The circuit shown in Figure 6.11 provides main memory addressing during the DMA operation. Two 8 bit data latches accept the upper and

lower bytes of the 16 bit address of the destination memory locations for the first PSD data byte. The address held in the data latches is fully software selectable.

The output of the data latches serve to preload a memory address counter which is incremented by the signal INCP. The counter outputs are buffered and placed onto the multibus when the signal $\overline{\text{TBUSY}}$ goes low.

6.5 The Test and Evaluation of the PSD Estimation Subsystem

6.5.1 Test System

The PSD estimation subsystem was constructed, debugged and then evaluated. The schematic diagrams reflect the design revision identified in the debugging process. The subsystem was evaluated using a test configuration shown in Figure 6.12. The PSD estimation subsystem was installed in and powered from the INTEL SBC-660 system chassis. Also included in the chassis was an INTEL SBC-80/20 single board computer and a SBC-116 combination memory and I/O board.

As implied by the diagram of Figure 6.12 the micro-computer permitted communication with the PSD, estimation subsystem and other peripheral elements. An operating system was developed for the 80/20 microprocessor to permit interaction with 1) the Silent 700 terminal to include tape playback and record, 2) the PSD estimation subsystem and 3) the TI 980A computer. Controlling software was developed and recorded on cassette tape. The Silent 700 terminal was employed to command the system operation and to enter the appropriate controlling software

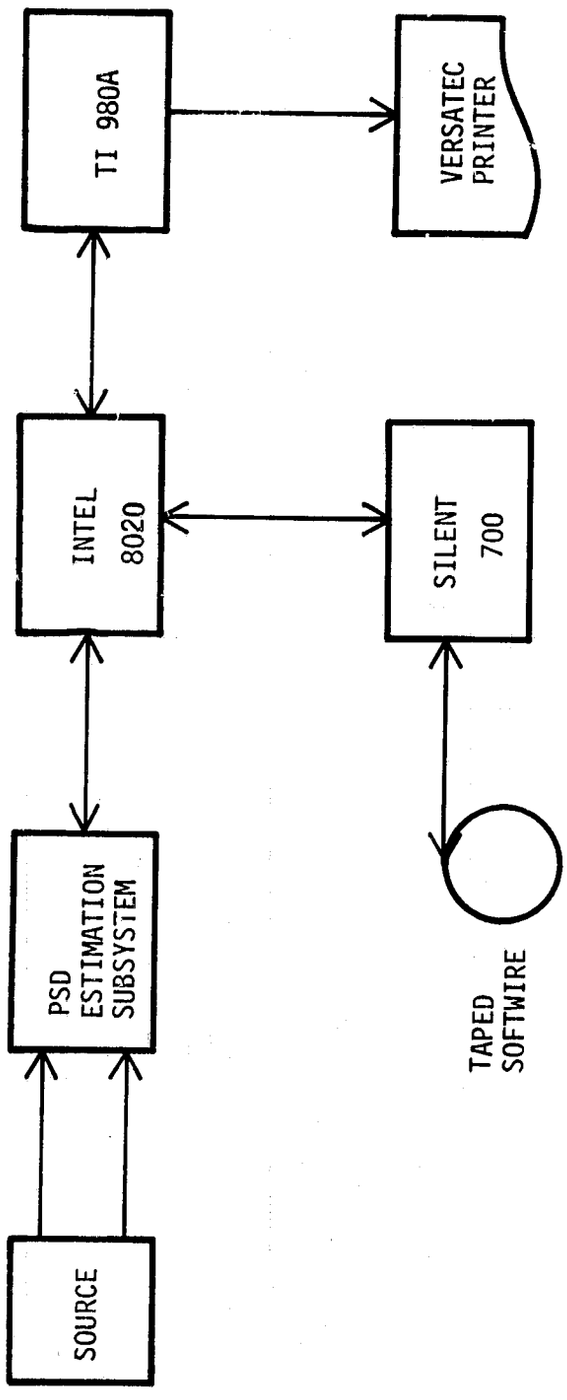


FIGURE 6.12 PSD ESTIMATION SUBSYSTEM TEST SET-UP

C-2

from tape. The software permitted interactive control from the Silent 700 and provided sufficient flexibility to display spectral data in hexidecimal on the local Silent 700 terminal or to transmit it to a remote TI 980A Computer for conversion to floating point and graphical display.

Various signals were injected into the test system to document the following parameters:

1) dynamic range
2) power linearity
3) frequency envelope
4) fore and aft signal separation

Other test results of interest were also generated. Prior to each sequence of test runs the CZT filtering unit was aligned in accord with the procedure outlined in Appendix C.

6.5.2 Test and other Illustrated Results

To test for the dynamic range and linearity of the PSD estimation system, a 2000 Hz tone was injected into the x channel with the input to the y channel grounded. The amplitude of the tone was varied over a 45 dB dynamic range. At each amplitude, 9 records were filtered, accumulated and displayed on the VERSATEC printer. Typical spectral plots for two different input amplitudes are illustrated in Figures 6.13 and 6.14. The peak spectial amplitudes were correlated with the input voltage represented in dB to demonstrate both the dynamic range and linearity of the subsystem. The result is shown in Figure 6.15. From this graph it is apparent that the subsystem exhibits a linear input power - out-

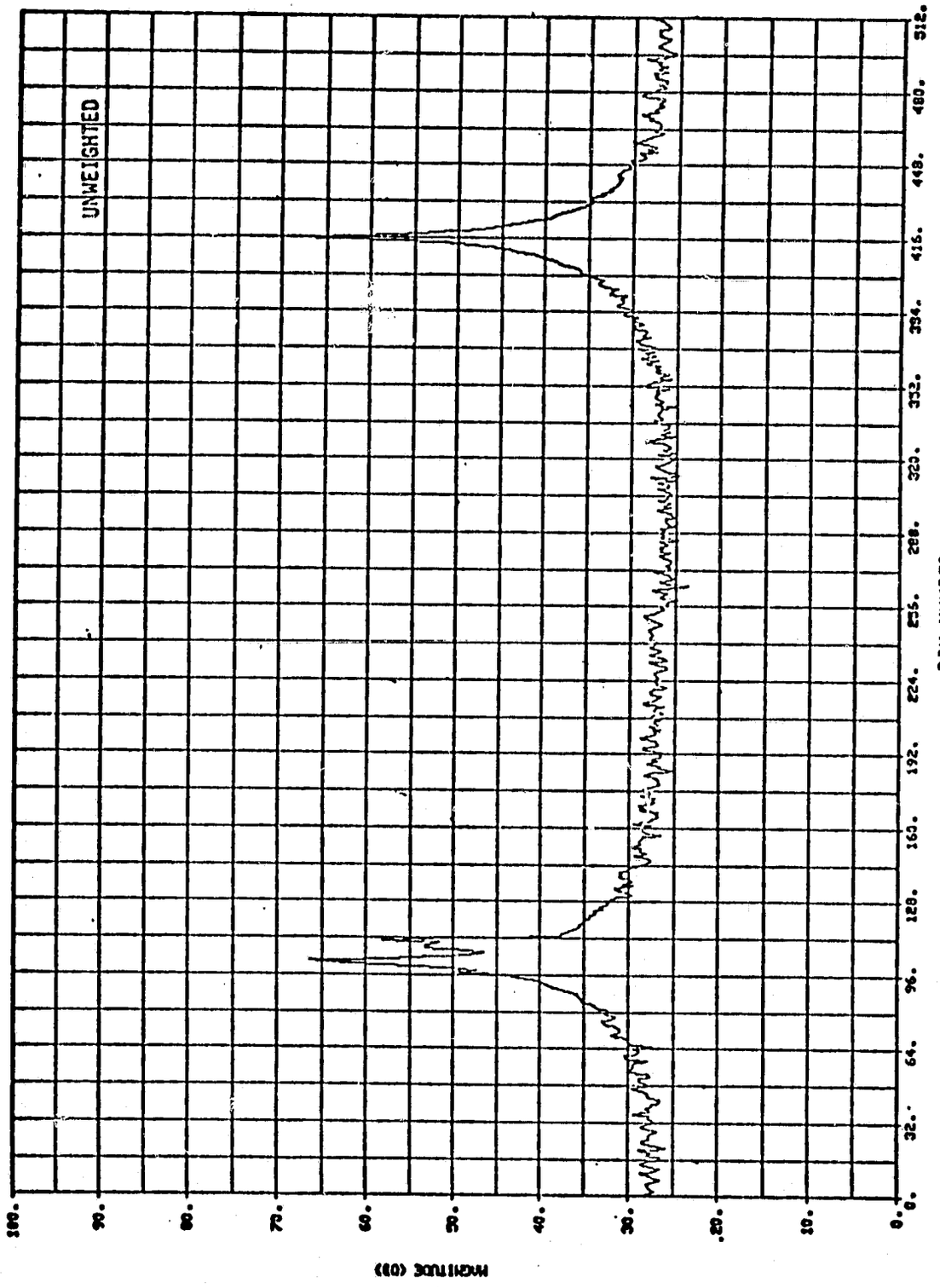


FIGURE 6.13 SPECTRUM OF 2000 Hz TONE WITH 1.0 VOLT RMS AMPLITUDE

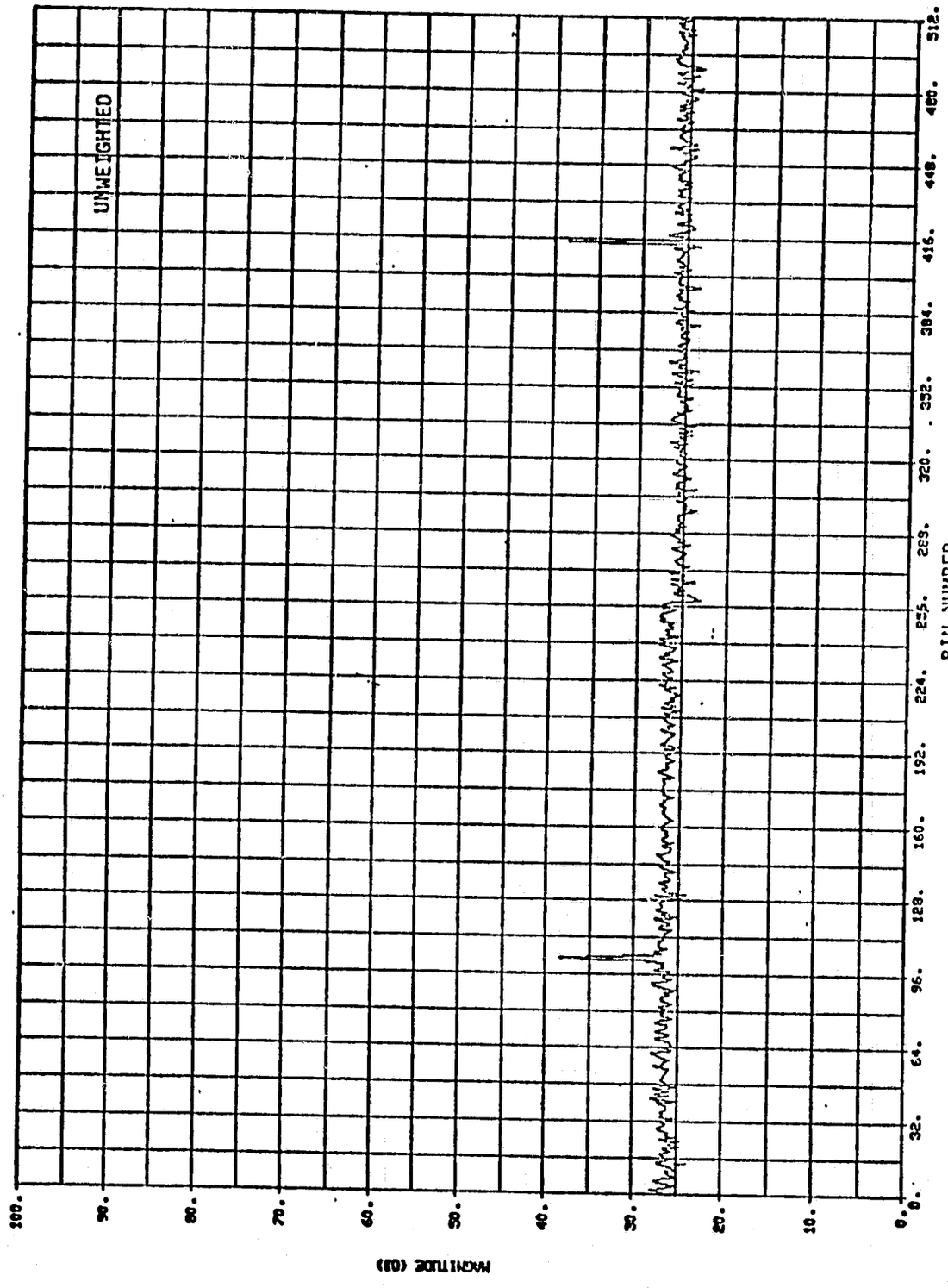


FIGURE 6.14 SPECTRUM OF 2000 Hz TONE WITH 0.03 VOLT RMS AMPLITUDE

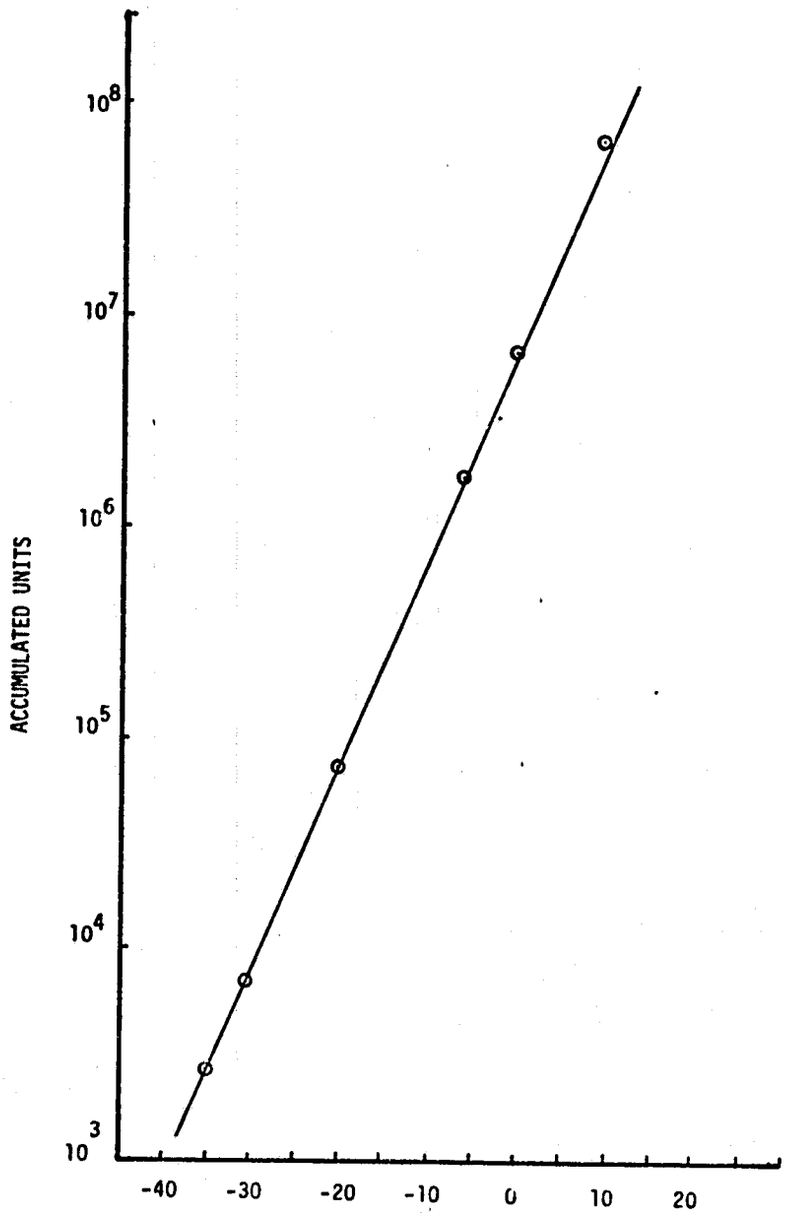


FIGURE 6.15 RELATIONSHIP BETWEEN "POWER" OUTPUT AND POWER INPUT

put power relationship over a 45 dB dynamic range. Since the noise floor is more than 10 dB beneath the peak spectial line for an input of -30 dB (see Figure 6.14), the actual dynamic range is somewhat greater than 50 dB with linearity undoubtedly occurring over this entire range.

The actual frequency response of a single Doppler filter (one of 512 parallel filters) was not measured because of the difficulty in maintaining a known frequency difference between the source and the Doppler filter center frequency. Visual experience from oscilloscope traces has, however, demonstrated that the unweighted RETICON filter has a sin X/X like response since when the source frequency matches a filter center frequency the side lobes essentially disappear. Similarly, where the Hanning weighted RETICON filter is employed, the first side lobes are 6 dB down when source and filter frequencies are aligned. The filter response envelope of the unweighted filter can be inferred approximately from the test data taken in the dynamic range and linearity test. Since the source and filter center frequency were misaligned, the adjacent spectial data represents the approximate response envelope of a single filter. The actual response will be within a few dB of that actually observed. The results of this test are illustrated in Figure 6.16. The graph represents a high resolution look of spectial lines of the type shown in Figures 6.13 and 6.14.

Since the NASA fan beam scatterometers illuminate the fore and aft track sectors (see Figure 3.1), both up and down Doppler spectra appear in the return. A well designed Doppler discrimination system must separate the fore and aft spectial data to assure isolation between

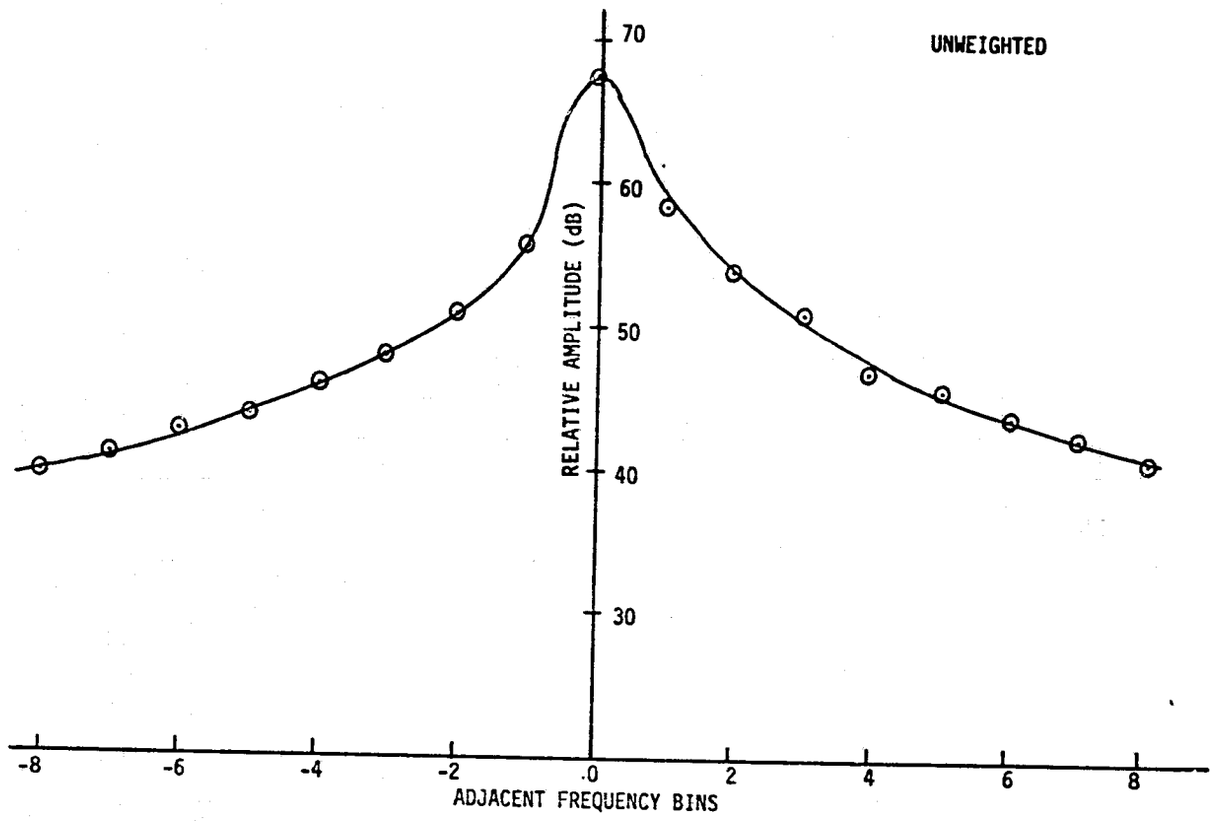


FIGURE 6.16 THE APPROXIMATE FREQUENCY ENVELOPE OF THE UNWEIGHTED FILTER

returns representing two azimuthal aspects of the distributive target. Test were conducted to determine whether the quadrature processing technique employed here exhibits sufficient isolation between fore and aft spectra. The isolation is largely a matter of properly balancing the gains in the quadrature channels.

To demonstrate the fore/aft isolation character of the CZT filtering unit, quadrature square wave signals were injected into the x and y input channels and the spectral information averaged over several accumulation cycles. As demonstrated in Appendix B, the spectrum of x+jy should be unsymmetrical with the 1st, 5th, 9th, etc. harmonics appearing on one side and the 3rd, 7th, 11th, etc. harmonics appearing on the opposite side of dc. The results of the these tests are illustrated by the graph of Figures 6.17 and 6.18 for the unweighted and Hanning weighted CCD filters, respectively. It is important to note that the spectra appear differently in the graphs for the two filters since their transversal taps are ordered differently. The dc spectral line for the unweighted filter appears on the left (it is actually displaced two bins to the right because of the delay injected by the S/H amplifiers); on the other hand the dc spectral line appears in the center bin (256) for the weighted filter (again actually 258 for the same reason). The fore/aft isolation may be inferred by comparing the amplitude of the 1st harmonic with the location in the negative spectrum where it would have occurred if it were a double sided spectrum. An examination of Figure 6.17 shows that the weighted CCD filter rejected the first harmonic in the negative portion of the spectrum (the righthand side) by more than 30 dB. Since

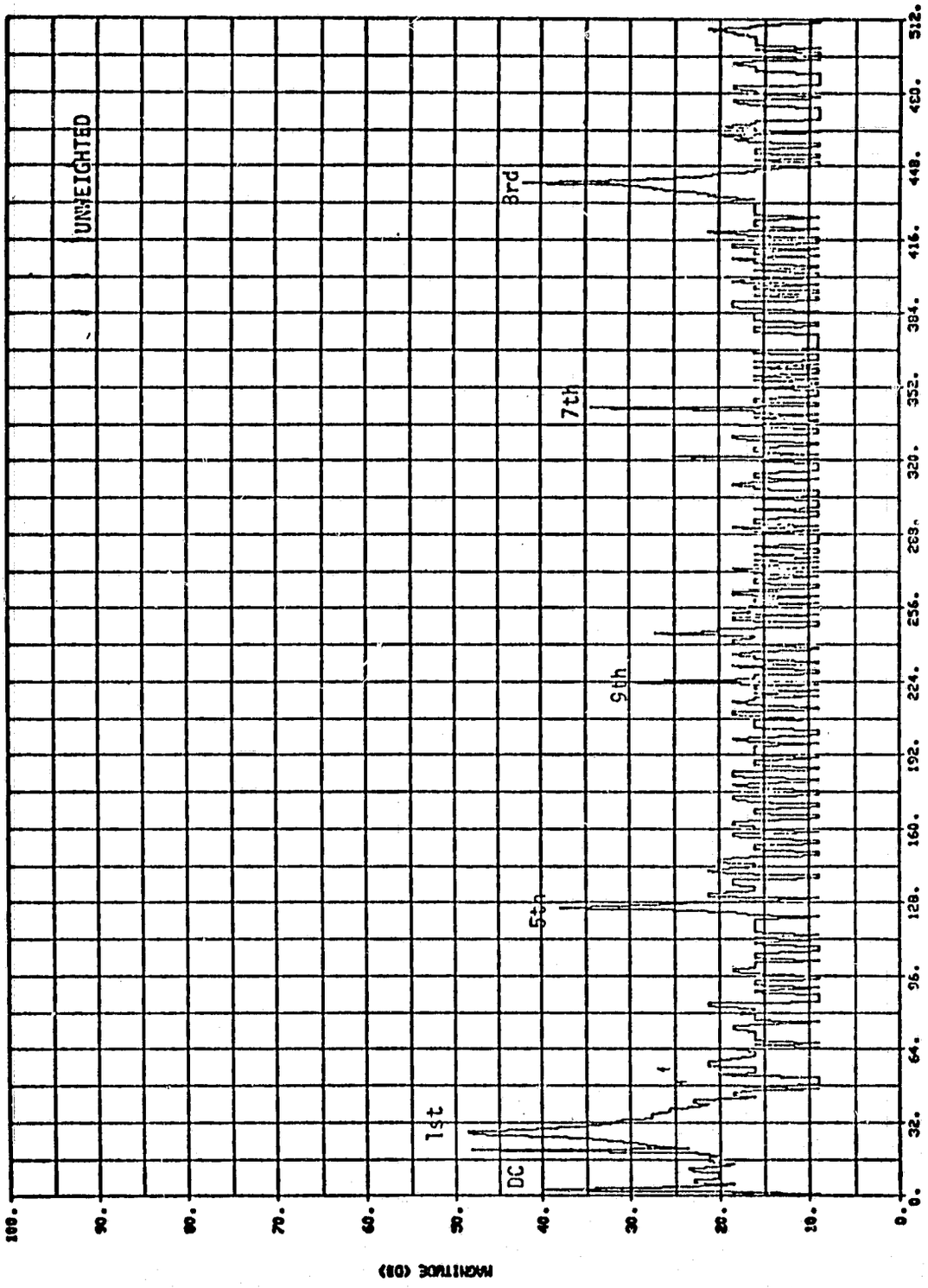


FIGURE 6.17 SPECTRUM 400 Hz QUADRATURE SQUARE WAVES (0.5 VOLT p-p AMPLITUDE)

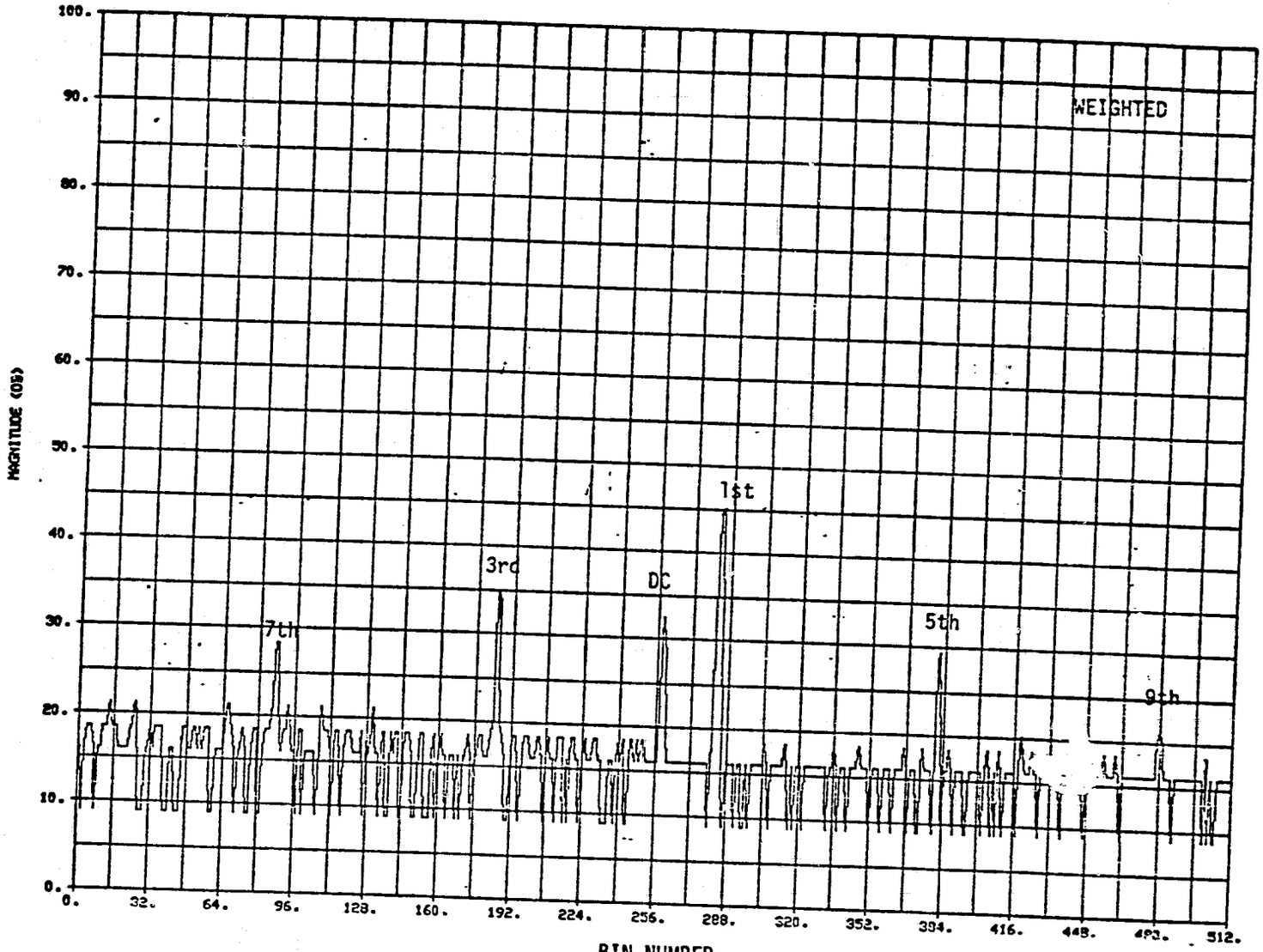


FIGURE 6.18 SPECTRUM OF 400 Hz QUADRATURE SQUARE WAVES (0.5 Volt p-p Amplitude)

the image of the first harmonic is not apparent on the righthand side, the amplitude of the test signal should have been increased to identify the actual isolation. But nevertheless 30 dB or more represents more than adequate isolation. An examination of the same spectrum produced by the weighted filter shows more than 25 dB isolation under the same input conditions. This result was anticipated since the gain through the weighted filter is 3 dB less than the unweighted.

Other test data were also generated to characterize the performance of the CZT filter unit on noisy signals. The graphs of Figures 6.19 through 6.21 represent PSD estimates of colored noise as generated by bandpass filtering broadband noise. Figures 6.19 and 6.20 show the affects of averaging 1 and 16 records, respectively, using the unweighted filter. Figure 6.21 represents a 16 record average through the weighted filter. Finally, the spectrum of actual C band scatterometer data was generated using the quadrature filtering technique. The result is shown in Figure 6.22. The 3000 Hz (approx.) calibration/polarization tone appears in the fore spectrum as it should. It is also noted that the aft and fore spectra are unsymmetric since the beam is primarily directed aft.

6.5.3 Discussion of Abberations in the Test Results

From this sequence of illustrations and actual experience several problems were identified in the PSD estimation system; however, all of the problems appear solvable by proper adjustments to the design, improving circuit fabrication practices for the analog board and reducing the background noise in the laboratory and its equipment. Identification

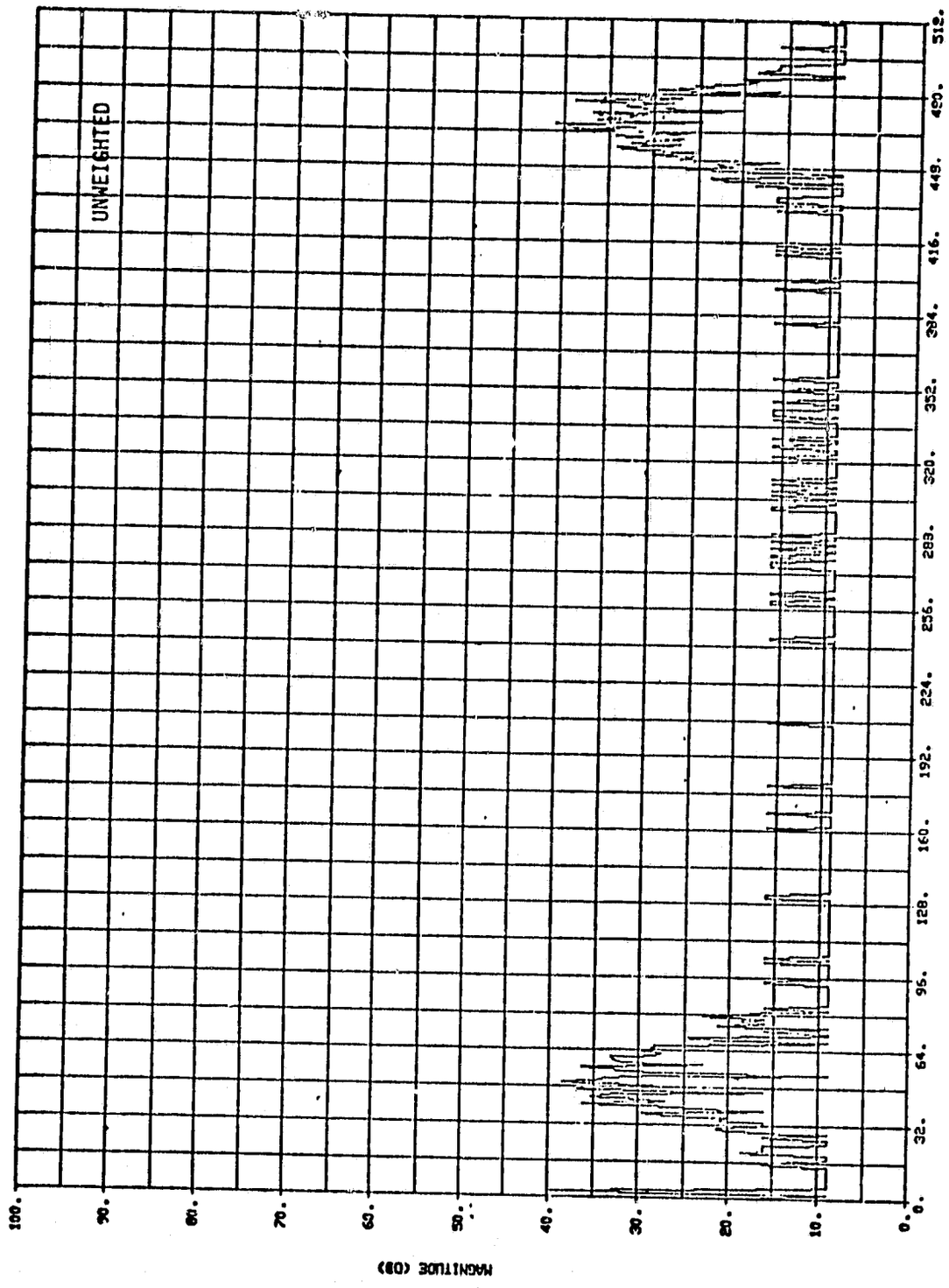


FIGURE 6.19 SPECTRUM OF COLORED NOISE CENTERED IN 1000 Hz (1 RECORD AVERAGE)

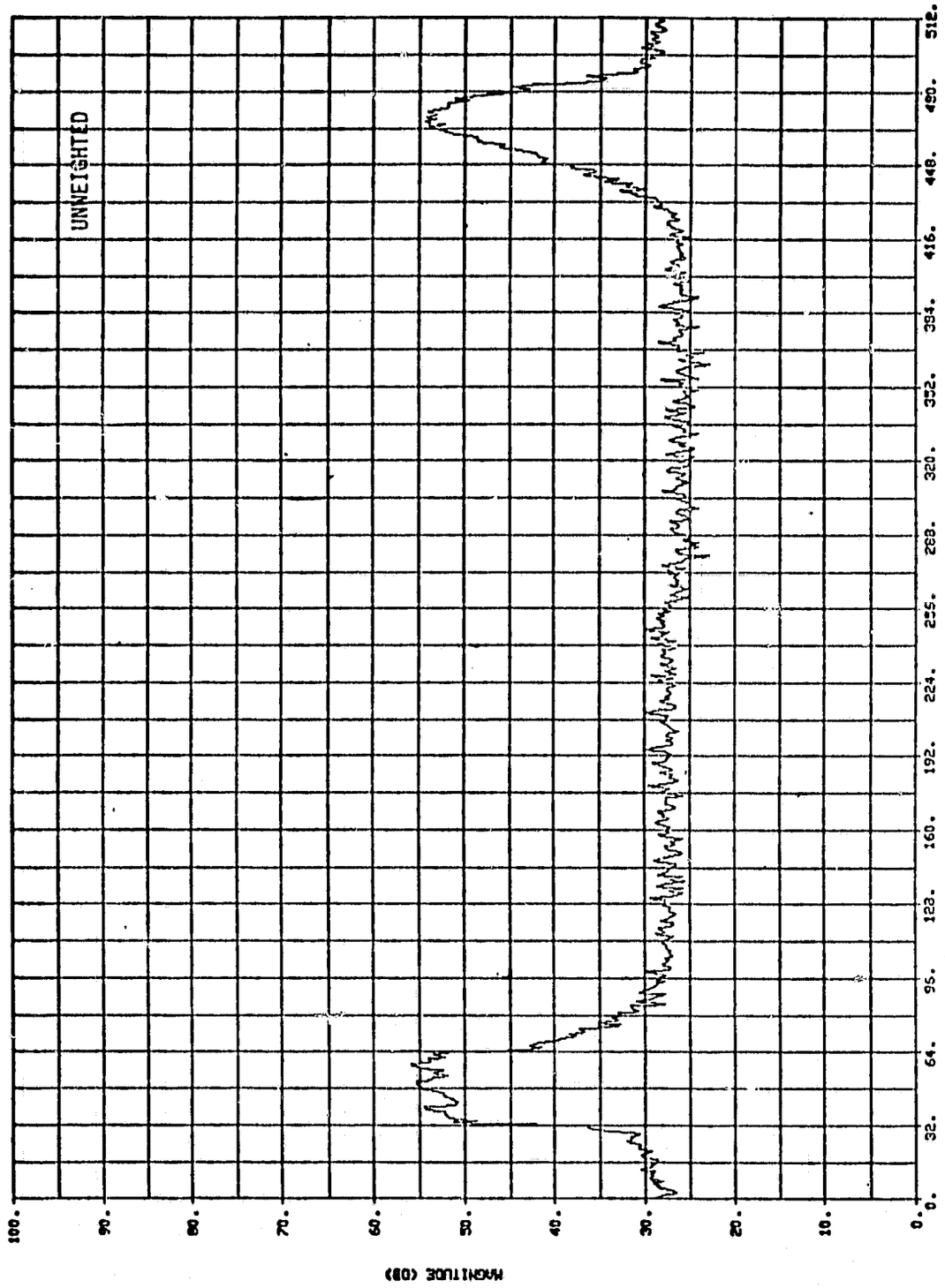


FIGURE 6.20 SPECTRUM OF COLORED NOISE CENTERED ON 1000 Hz (16 RECORD AVERAGE)

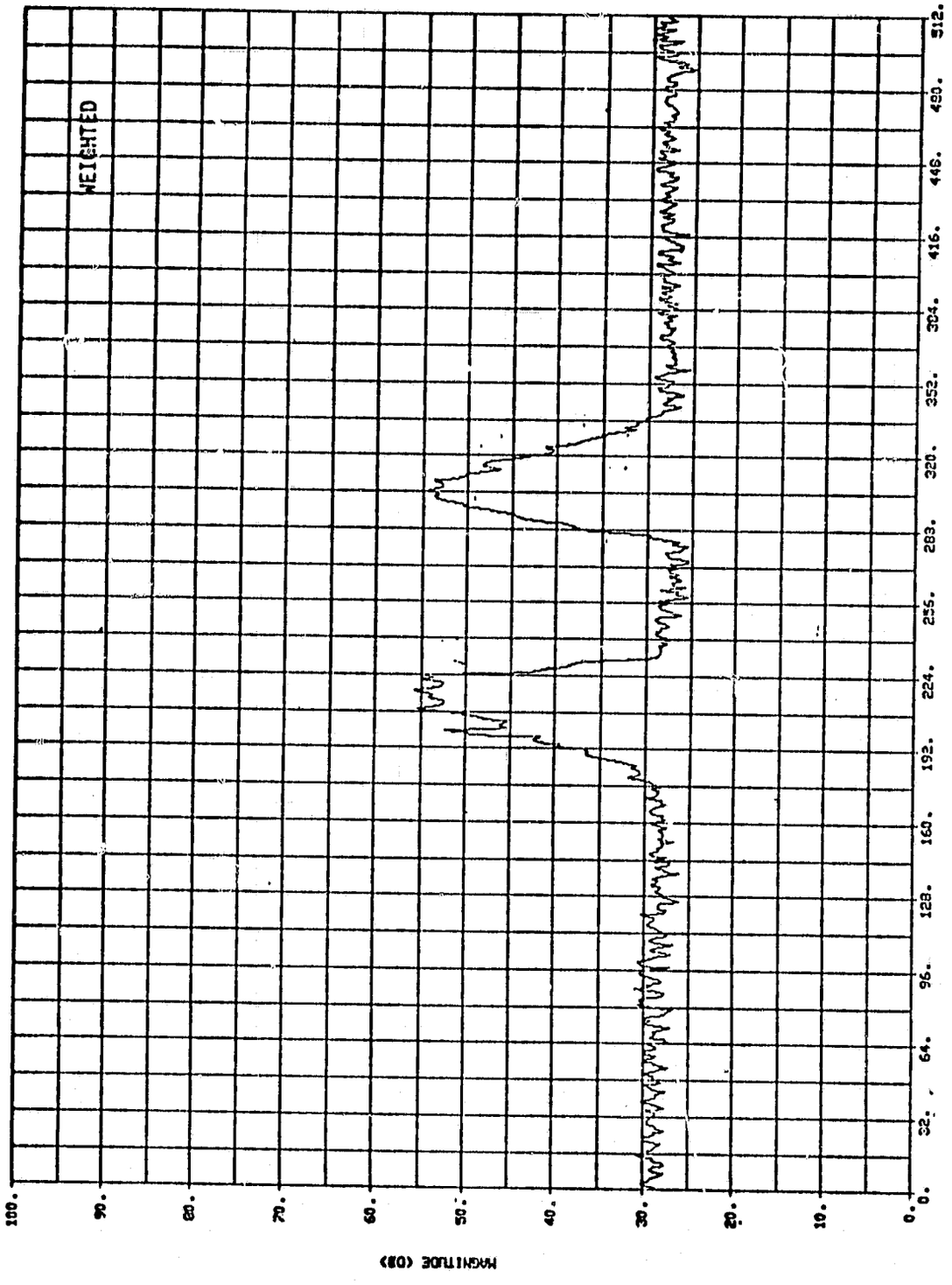


FIGURE 6.21 SPECTRUM OF COLORED NOISE CENTERED ON 1000 Hz (16 RECORD AVERAGE)

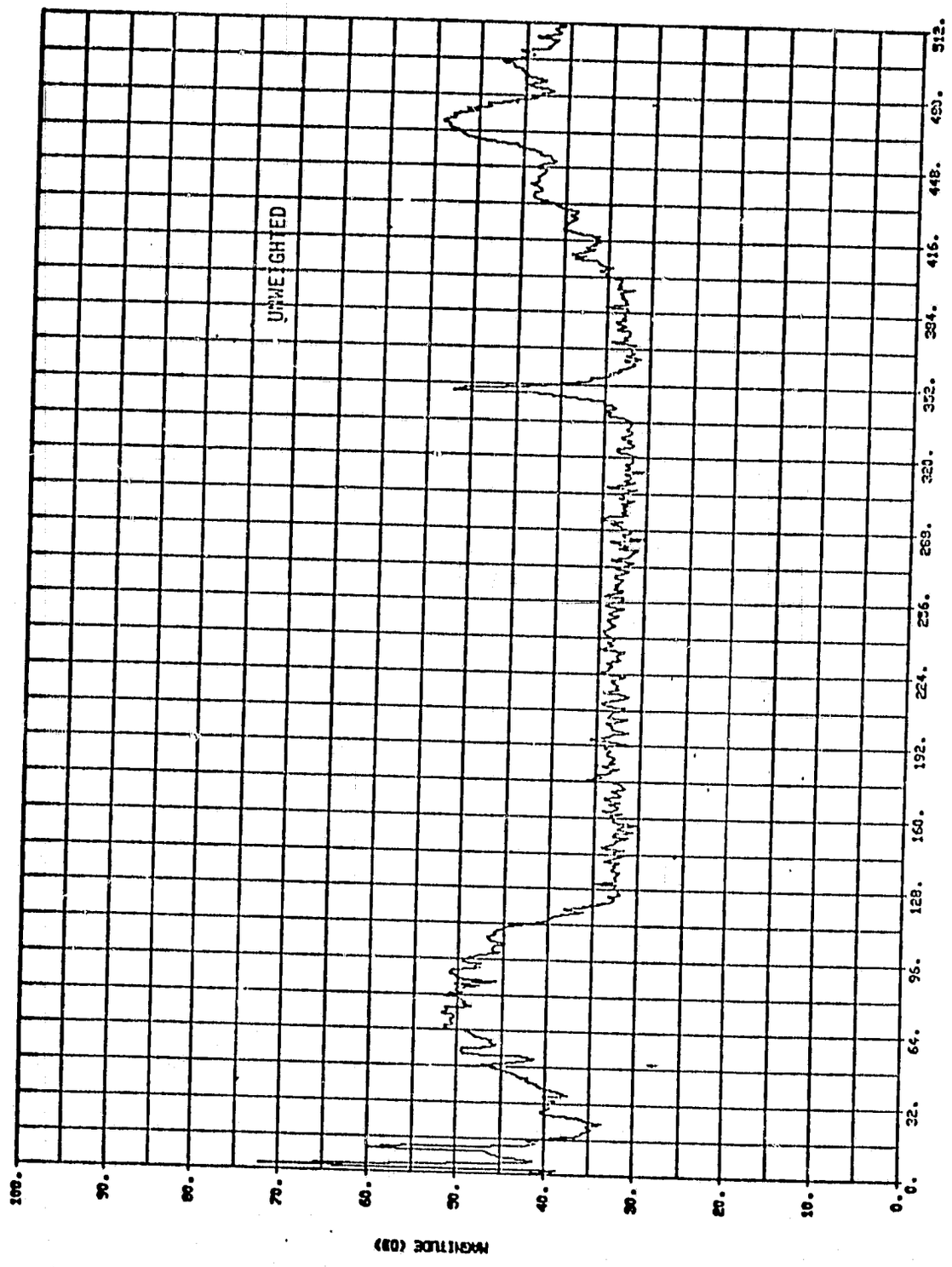


FIGURE 6.22 SPECTRUM OF A C-BAND SCATTEROMETER RETURN SIGNAL

and discussions if these problems are treated in the following paragraphs.

An examination of the graph of Figure 6.19 suggests that the quantum jumps in the spectra between the major noise lobes is not associated with the CCD device or the input noise. These constant amplitude jumps may be attributable to a high noise environment for the A/D converter induced by using wire wrapping techniques in a high density digital circuit. The problem is thought to be a major contribution to the high noise floor level in all of the spectral illustrations produced by averaging several records. It is suspected that these quantum jumps may be eliminated by mounting the A/D converter on doubly clad circuit board to shield its terminals from the digital and laboratory noise.

An examination of Figure 6.23 shows a spurious spectral response on the right of the 2000 Hz line. The source of the spurious response is unknown, but it is known to occur where high amplitude signals, not exceeding the dynamic range of the CCD device, are injected into the filtering unit. The spurious response primarily occurs on the positive frequency side of the spectrum and seldom occurs over the negative frequency spectrum. This property is thought to induce the lack of symmetry in the positive and negative spectrums of Figures 6.20 and 6.21. The fact that it primarily occurs over the positive frequencies and not the negative frequencies strongly suggests that the problem is solvable.

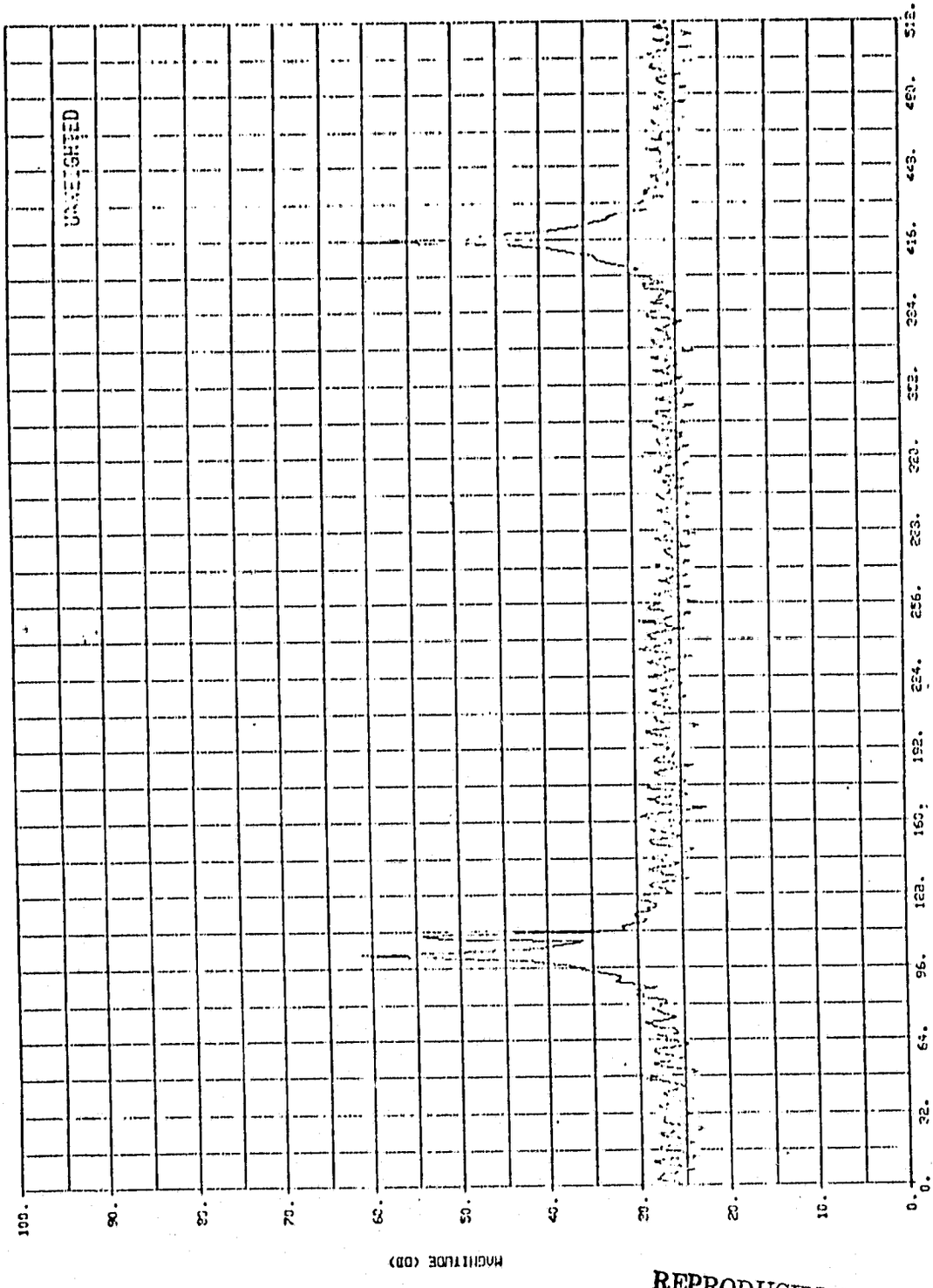


FIGURE 6.23 SPECTRUM OF 2000 Hz TONE WITH 0.5 VOLT RMS AMPLITUDE

7.0 THE MICRO-PROCESSING SUBSYSTEM AND INTERFACING UNITS

7.1 Microprocessor

7.1.1 Multibus System Description

The signal processing system consists of the following interconnected units:

1) Front panel and operator interface
2) Aircraft interface
3) Output interface
4) CZT filtering unit
5) PSD and DMA controller unit
6) Detection and accumulation unit
7) CPU
8) Memory

It is widely accepted that the most efficient way of establishing the necessary communication paths between these units is through the use of a system interconnection bus [8].

The Intel Multibus was selected for this application because of its industry-wide support and its flexibility. The Multibus includes the following signal lines:

16 address lines
16 bi-directional data lines
8 multi-level interrupt lines

The address and data lines are driven by three-state devices, while the interrupt and some other control lines are open-collector driven.

Modules using the Multibus have a master-slave relationship. A bus master can command the bus, while bus slaves cannot. The CPU is a bus master in this application, while the memory is a bus slave. The Multibus is expandable to a maximum of 16 masters.

The system also has a DMA controller which acts as a bus master. Bus arbitration results when the DMA controller requests use of the bus while the CPU controls it. The bus clock, derived from the CPU, provides a synchronous reference for the resolution of bus arbitration. The Multibus provides for a maximum data transfer rate of 5MByte/second. System use of the multibus is found in Section 6.4 and in vendor's literature describing the CPU and the memory [9].

7.1.2 System Components

7.1.2.1 CPU

Early in this contract the 8080 family of microcomputer was selected as the unit best suited for this application [10]. It was recommended that the CPU, memory and available I/O be purchased rather than constructed to take advantage of vendor reliability and maintenance support. Since the Intel SBC-80/20 was introduced in 1976, other vendors, including Christian Industries, National Semiconductor, Monolithic Systems, Advanced Micro-Computers, MuPro, and even Intel (with their SBC-80/30) have introduced advanced products to dramatically extend the capabilities of the well designed Multibus system. For example, Advanced Micro Com-

puters has introduced their Multibus compatible AMC 95/4000. Both the SBC-80/20 and the 95/4000 fulfill system requirements for 8080 compatibility, on board RAM and EPROM, and programmable interrupt structure (see Figure 7.1). But the 95/4000 has two features which make it uniquely superior to the 80/20.

The first is processor speed. The 95/4000 comes in two versions, the faster of which contains a 4 MHz 8080. This means that the 95/4000 can execute instructions twice as fast as the SBC-80/20. The second superior feature of the 95/4000 is its on-board Am9511 Arithmetic Processing Unit. It allows for the evaluation of ordinary and transcendental mathematical functions up to 100 times faster than software [11].

The advantages of the 95/4000 over the 80/20 may be quantified by examining the effect of the faster processor speed and arithmetic processor on program execution speed and memory requirements.

Program execution time is divided between mathematical software and control software. Based on an analysis of the 13.3 GHz scatterometer processor software, which is assumed to be similar to that of the proposed C- and L-band processors, it was found that the processor spends about 93% of its time doing mathematical software and only 7% doing control functions. If T is the total execution time for the program this may be written as

$$T = 0.93T + 0.07T \tag{7.1}$$

where 0.93T is the time due to the execution of the mathematical software and 0.07T is the time due to control functions. If these times could

	80/20	95/4000
Word size	8 bits	8 bits
On board RAM	2K bytes	2K bytes
On board EPROM	0/8K bytes	0/12K bytes
Addressable memory	64K bytes	64K bytes
Clock frequency	2 MHz	2/4 MHz
I/O parallel	48 lines	48 lines
I/O serial	1 RS-232C	1 RS-232C and 1 TTY
Arithmetic capability	No*	Yes
DMA capability	No**	Yes
Priority interrupt	Yes	Yes

*available with the addition of the SBC-310

**available with the addition of the SBC-501

FIGURE 7.1 COMPARISON OF SBC-80/20 and AMC 95/4000

be decreased independently by α and β, respectively, then the new execution time is

$$T_{NEW} = 0.93\alpha T + 0.07\beta T \qquad (7.2)$$

where α = $\frac{\text{time to do math at increased speed}}{\text{time to do math software without increased speed}}$

and

β = $\frac{\text{time to do control software at increased speed}}{\text{time to do control software without increased speed.}}$

Using the 95/4000, $\alpha = 0.109$ and $\beta = 0.667$ (see section 7.1.2.2). As a result, the new execution speed is

$$T_{NEW} = (0.93)(0.109)T + (0.07)\ (0.667)T = 0.148T \qquad (7.3)$$

New execution time is seen to be 85% less than the old. This means an increase in execution speed by a factor of almost 7. Concern has been expressed that the 4 MHz 95/4000 would be difficult to use with a microprocessor development system due to its increased clock rate. If we use the 2 MHz 95/4000, then $\beta = 1.0$ and α remains at 0.109. The impact on execution speed of using the 2 MHz 95/4000 is then

$$T_{NEW} = (0.93)\ (0.109)T + (0.07)(1.0)T = 0.171T \qquad (7.4)$$

This corresponds to a decrease in execution time of 83% or an increase in execution speed by a factor of almost 6. The conclusion is that even with a "slow" 95/4000, execution speed is increased by almost 600% due to the large increase in speed of the mathematical functions.

Similarly, memory usage may be divided between that used to store the mathematical software and that used to store the control software and data. For the 13.3 GHz processor, this may be written as

$$M = 0.678M + 0.322M \tag{7.5}$$

where 0.678M is storage for non-mathematical software and 0.322M is memory for the mathematical subroutines. The only portion which we of a 95/4000 would decrease is the mathematical routines. The new equation is

$$M_{NEW} = 0.678M + 0.322\gamma M \tag{7.6}$$

where γ = $\frac{\text{memory required for math routines without 95/4000}}{\text{memory required for math routines with 95/4000}}$

The analysis show that γ can be as small as 0.116. Thus, the new memory requirement is

$$M_{NEW} = 0.678M + (0.322)(0.116)M = 0.715M$$

The use of the 95/4000 decreases memory requirements by 28.5%. The conclusion is that the arithmetic processor makes a significant difference in both execution speed and memory requirement.

When the above analysis are considered, it is concluded that because the arithmetic processor offers such a significant advantage, it would enhance the capabilities of <u>any</u> Multibus compatible processor module. In view of the experience gained with the SBC-80/20 and of the relative costs ($1430 for the 80/20 plus the AMC 95/6011 APU versus

$1390 for the 2 MHz 95/4000) it was decided that the purchase of the 95/6011 APU and the SBC-80/20 would best satisfy present and anticipated system requirements.

A complete description of the Intel SBC-80/20 and the AMC 95/6011 is found in the vendor's literature for each part.

7.1.2.2 Memory

The software system required 50K bytes of RAM and 14K bytes of EPROM for its operation. Since the 80/20 provides 2K bytes of RAM and sockets for up to 4K of EPROM, an addition of 48K bytes of RAM and sockets for 10K of EPROM is required. Intel provides 48K of Multibus compatible RAM with their SBC-416. The memory map of the processing system as implemented is shown in Figure 7.2. A complete description of the SBC-048 memory is found in the vendor's literature.

7.1.2.3 Front Panel and Operator Interface

The front panel is interfaced as an I/O device to the microprocessor subsystem. It is wire-wrapped and therefore requires two card slots. Its design and operation are described in section 7.2.

7.1.2.4 Aircraft Interface

The interface to the aircraft NERDAS computer is constructed on a wire-wrapped card and requires two slots. It is fully described in section 7.3.

7.1.2.5 Output Interface

The interface to the aircraft Bi ϕ-L tape recorder is constructed on a wire-wrapped card and requires two slots. It is documented in section 7.4.

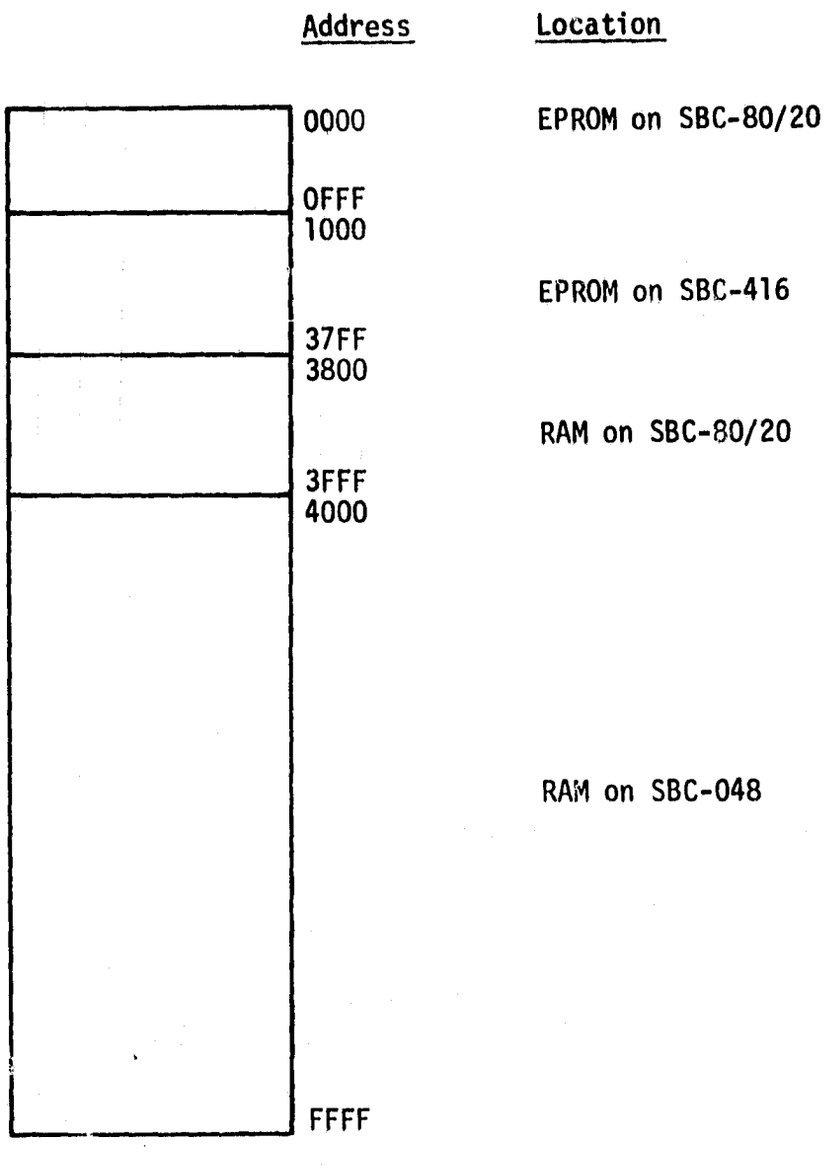


FIGURE 7.2 SYSTEM MEMORY ORGANIZATION

7.1.2.6 PSD Estimation Sub-system

The PSD Estimation Sub-system is constructed on three wire-wrapped circuit cards and requires six slots. It is described in section 6.4.

7.1.3 System Chassis and Power Requirements

7.1.3.1 System Chassis

The system as described above requires 13 card slots. The MDS-800 chassis from Intel was first considered as a potential system chassis. However, although the MDS-800 has enough slots to contain the single-polarization development system, a dual-polarization system would require more than 16 slots. Thus, a MDS-800 system is not upward compatible. To satisfy potential system expansion requirements, an expandable chassis consisting of intially one SBC-604 and three SBC-614's (all from Intel) provides 16 slots of chassis space, expandable to 20 by the addition of a single additional SBC-614. A tentative card slot arrangement is given in Figure 7.3. Information on the SBC-604 and -614 is found in the vendor's literature.

7.1.3.2 System Power

The system power requirement is given in Figure 7.4 and is satisfied through the use of properly sized fixed voltage power supplies. However, greater flexibility allowing the adjustment of analog voltages to minimize noise is provided if the supplies for items (4) and (5) are made variable. Connection of the power supplies to the system bus is as follows:

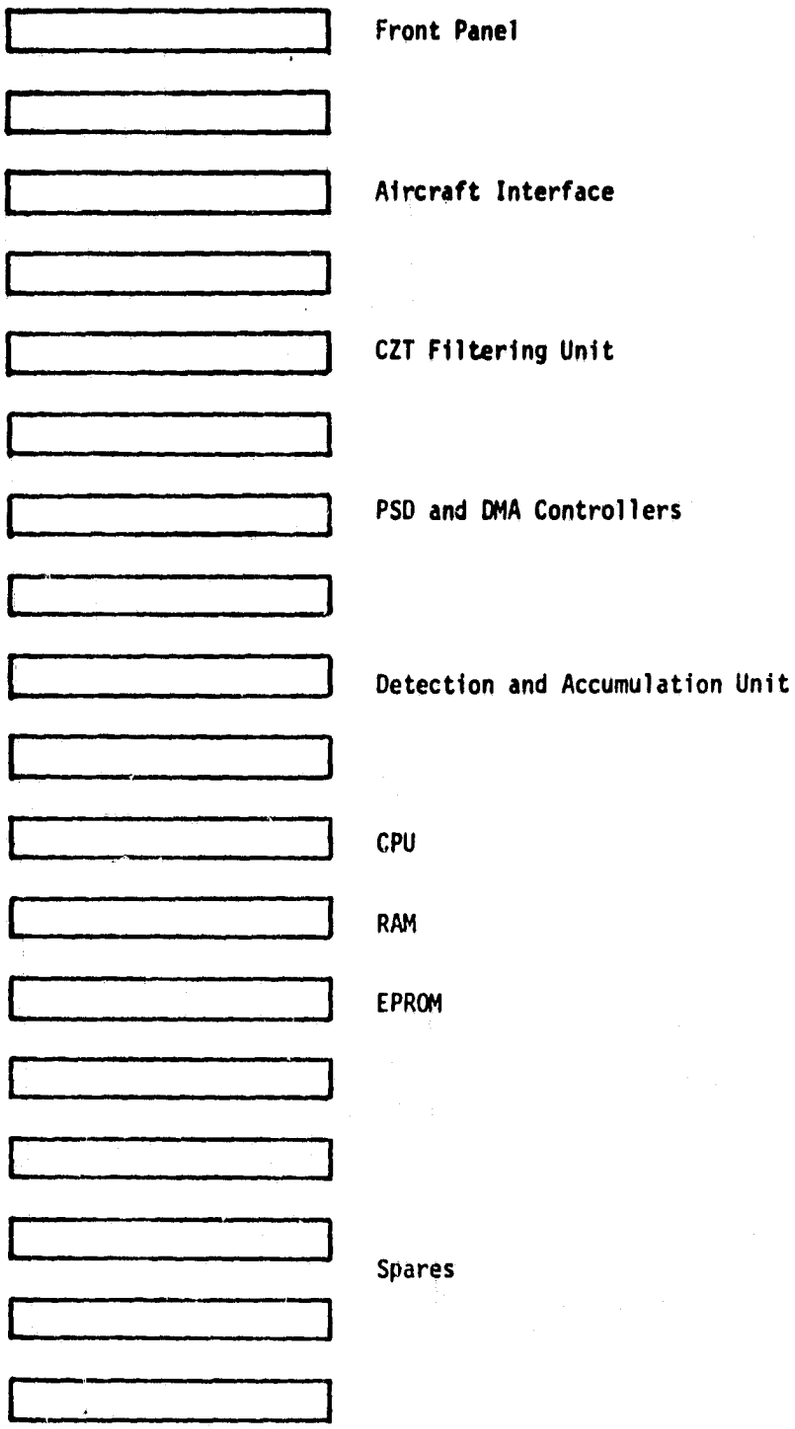


FIGURE 7.3 CARD SLOT ASSIGNMENT

Voltage	Current Requirement
+5	<30 A *
-5	200 mA
+12	360 mA
-12	25 mA
+16	350 mA
-16	200 mA
+20	25 mA

*Note: 30A is rated output of power supply; actual requirement is less than 30A.

FIGURE 7.4 POWER REQUIREMENTS

Multibus Pin	Definition
3, 4, 5, 6, 81, 82, 83, 84	+5
7, 8	+12
79, 80	-12
28	+16
30	-16
34	+20
1, 2, 11, 12, 27, 29, 32, 75, 76, 85, 86	GND

7.2 Front Panel and Operator Interface

7.2.1 Design

The contract requires an integral front panel allowing the operator to over-ride NERDAS aircraft parameters in the event of aircraft equipment failure during a mission. A design meeting this requirement is shown in Figure 7.5. The central part of the design is two Matrox MTX-A1 programmable keyboard/display controller chips [12]. The front panel (Figure 7.6) contains a 16 key keyboard for the entry of over-ride data and 4 keys for control of machine state (see section 8). The top 11 display positions are for displaying the over-ridden parameters and their values. Over each of the bottom nine displays is a legend. The four on the left show an '*' when the calibrate signal is absent, when the system's data buffer is filled, or when the integration time is inadequate. On the right most five displays, a blank is displayed if a parameter is not over-ridden; otherwise, an '*' is displayed under the legend corresponding to the over-ridden parameter. Complete parts

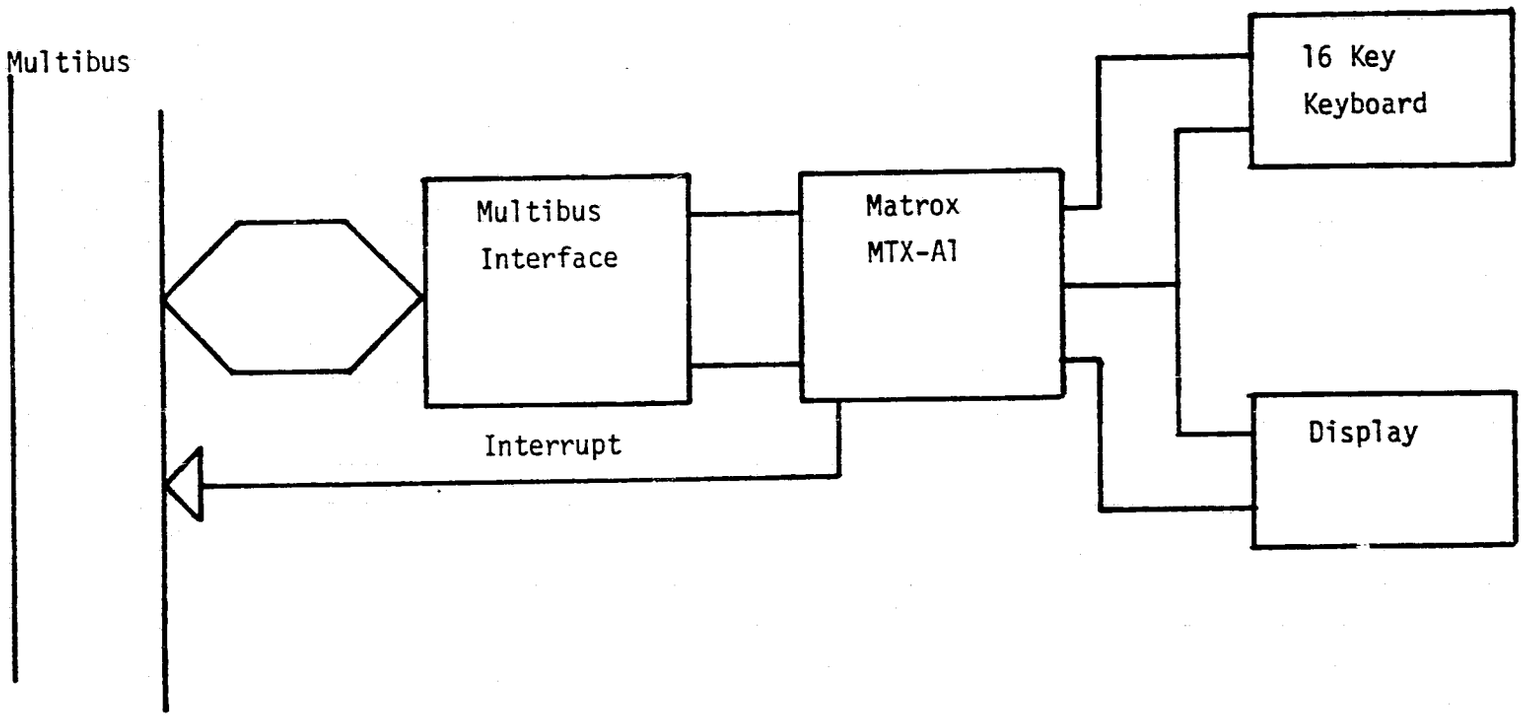


FIGURE 7.5 OPERATOR INTERFACE AND DISPLAY

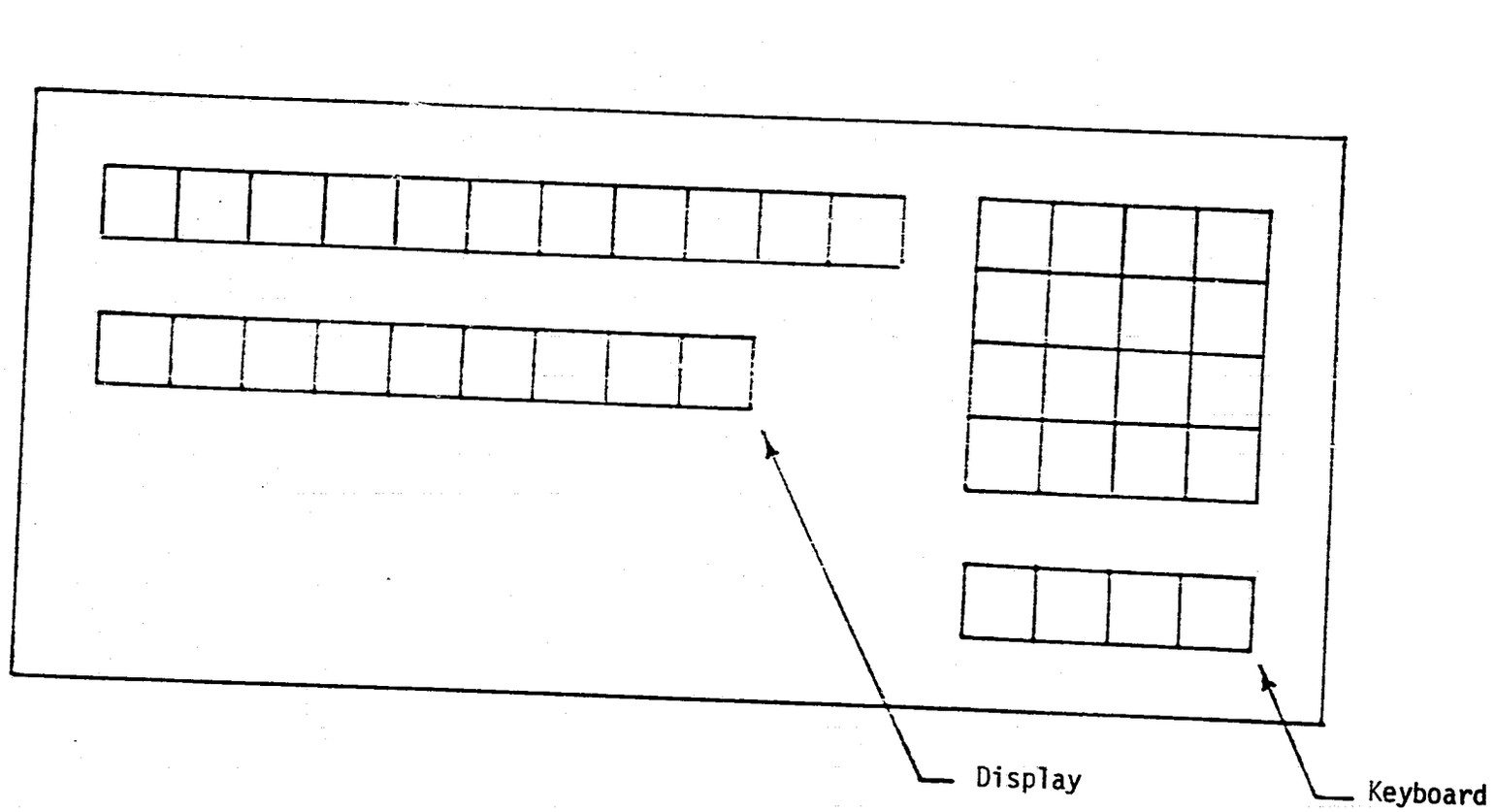


FIGURE 7.6 SYSTEM FRONT PANEL AND OPERATOR INTERFACE

lists, vendors, and schematics are given in the appendices. The design is otherwise self-explanatory.

7.2.2 Overview of Usage

The following addresses are used by the microprocessor for control of the CZT hardware and the front panel display.

Address (hex)	Function
08	LOADN/
09	DMA/
0A	LOADLSB/
0B	LOADMSB/
0C	DISP1A/
0D	DISP1B/
0E	DISP2A/
0F	DISP2B/

These functions are described below in terms of their affect on the hardware. A later section details the software operation.

DISP1A/ references the first of the two Matrox MTX-A1 keyboard/display controllers. Data may be written to positions 1-6 of the top display and positions 1-9 of the bottom display (see Figure 7.6) with an OUT instruction or read from the keyboard with an IN instruction. A read from DISP1B/ is used to load the contents of the status register of the first MTX-A1 into the microprocessor accumulator.

DISP2A/ references the second MTX-A1. Data may be written to positions 7-11 of the top display using an OUT instruction. An IN instruction from this address produces unpredictable results. DISP2B/ is used to load the contents of the status register of the second MTX-A1 into the microprocessor accumulator.

The operation of the panel has been optimized for user convenience. Suppose that the NERDAS altitude was to be over-ridden. The desired numeric value is entered via the keyboard, ENTER is pressed, and the alternate function of the 8 key is selected for ALTITUDE. The DISP function allows the operator to display the over-ride value for a particular parameter which has been over-ridden. The functions of the CLEAR, RUN, HALT, and RESET keys, as well as further details of front panel operation, are available in Section 8.

7.3 Aircraft Interface

The aircraft interface module provides the capability to extract the aircraft flight parameter data that is required for σ° computation from the NERDAS Bi-ϕ L bit stream. There are three basic functions required in the data extraction operation. The first step is to generate a phase reference and master clock signal from the NERDAS data signal. This operation is accomplished by applying the input signal to a full wave rectifier to remove the encoded phase information. The resulting signal serves as the input to a phase locked loop (PLL) circuit which provides a stable phase reference signal at the precise frequency of the input signal.

The second step in the data extraction operation is to decode the incoming data stream. This decoding operation consists of detecting the encoded pattern of ones and zeros from the Bi-ϕ L signal and generating a corresponding TTL level data sequence. The TTL sequence is continually examined to identify the presence of a 12 bit sync pattern that signifies the beginning of a frame of aircraft flight parameter data. When the beginning of a data frame is detected the serial bit pattern is synchronously accepted by one of the serial interface ports of the SBC 80/20 microprocessor. The port converts the serial data to 8 bit parallel data such that each data byte corresponds to two 4 bit NERDAS BCD digits. The data are stored directly in the processor memory and are immediately available for use in the σ° computation.

By using one of the serial interface ports of the SBC 80/20 the design of the NERDAS interface is considerably simplified in terms of both hardware and software. The acquisition of NERDAS data is fully controlled by software and data transfer occurs only when expressly requested by the σ° computational algorithm.

7.4 The Bi-ϕ L Interface

The output interface of the real-time scatterometer processor is a 100 KHz Bi-ϕ L serial data channel which will be recorded on a 14 track analog magnetic tape unit. The output interface is very similar to that employed in the 13.3 GHz real-time processor [1].

The Bi-ϕ L output frame consists of 256 4-bit data words. The first three of these words are frame SYNC words. Following the frame

SYNC words is a frame identifier word which indicates the type of data contained in the remainder of the frame. A data frame may either contain calculated radar cross section data or it may be a fill frame which consists of alternating 1's and 0's.

Figure 7.7 is a block diagram of the Bi-ϕ L output interface. Three 8212 data latches contain the SYNC and frame identifier words which form the beginning of each data frame. Control logic is provided to select the proper frame identifier word at the beginning of each output frame.

A universal synchronous receiver transmitter (USRT) performs the parallel to serial conversion required by the Bi-ϕ L generator. A bit counter is provided to signal that 256 words have been transmitted and that a new frame must begin.

Transfer of the σ° data from the CPU to the Bi-ϕ L interface is made via the SBC-80 parallel interface port.

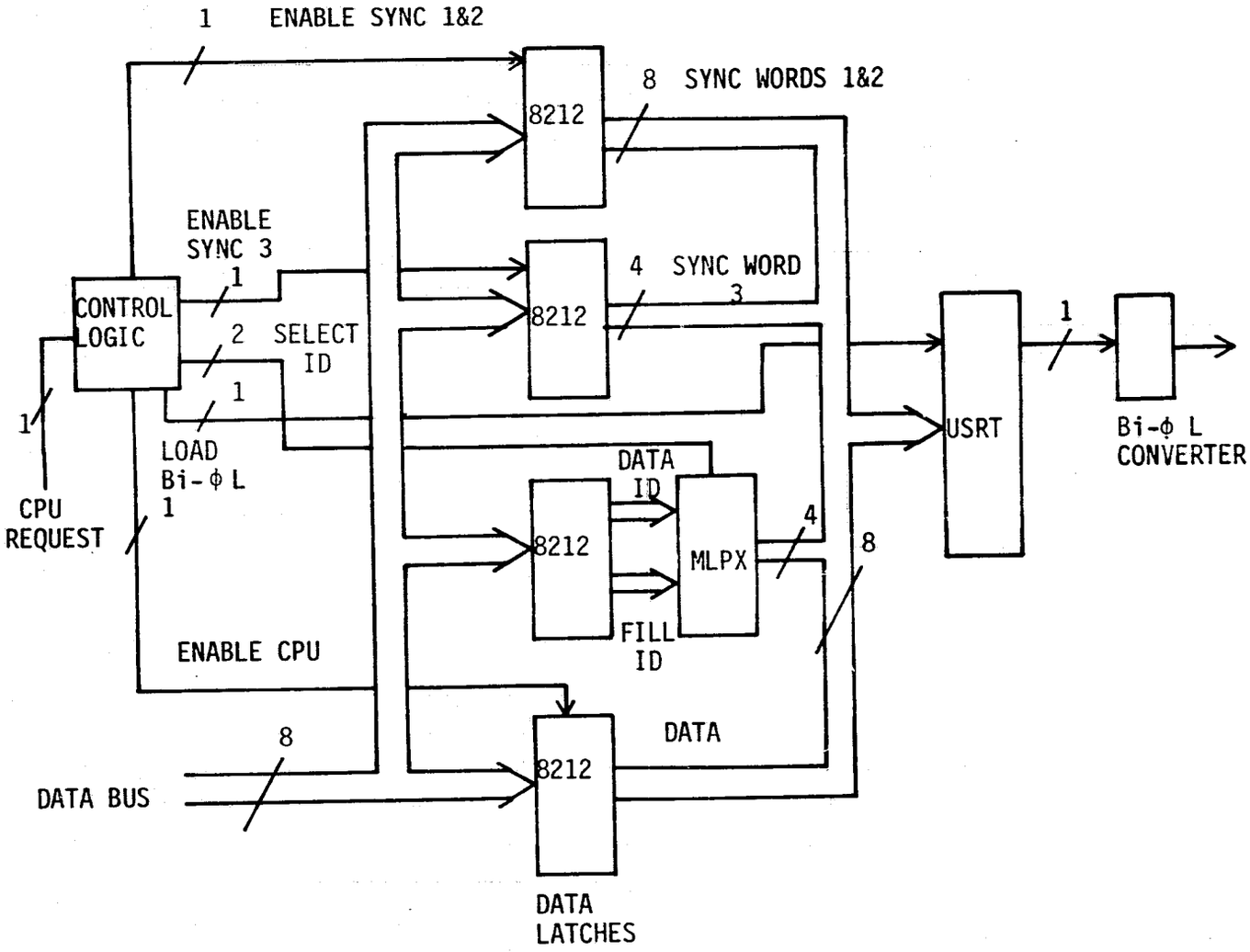


FIGURE 7.7 BLOCK DIAGRAM OF BI-ϕ L OUTPUT INTERFACE

8.0 SYSTEM SOFTWARE

8.1 Design Rationale

The overall software system is designed around two major functions; i.e., 1) system operation and control, and 2) the scattering coefficient calculations. The first, system control, constitutes performance of those software functions related to the "HALT" and "RESET" modes. This software handles all operator interactions needed to specify conditions under which the system will "RUN". Specifically, this means identifying aircraft data values to be over-ridden, if any, and what override values to use.

While in the "RESET" mode the software system will accept from the operator those specific over-ride values of the aircraft data system (NERDAS) to be used in the sigma-zero calculations. It will also display current contents of these over-ride values.

Minimum operator interaction has been the over-all guiding criterion in the software design. For example, if NERDAS data does not need to be over-ridden the operator needs only to "RESET" the system, select the appropriate Receiver Polarization, and press "RUN". The software system will perform all other required functions necessary to generate the output data frames.

In discussing the software functions for scattering coefficient calculations it is appropriate to first review the system geometric concepts.

8.1.1 System Geometry and Correction Factors

The basic objective of the micro-processor software is to calculate, repeatedly, the estimated radar scattering coefficient, $\hat{\sigma}^o$, using timing and position data input from the aircraft data system and radar transmit/receive power information from the chirp Z-transform hardware. This is done by implementing the radar equation in its general form

$$\hat{\sigma}^o = \frac{(4\pi)^3}{\lambda^2} \frac{R^4}{A} \frac{Z(f)}{K} \frac{PR}{G_r G_t} \tag{8.1}$$

where R = range to the target ground cell

A = area of the target ground cell

Z(f) = receiver roll-off filter correction

PR = receiver/transmiiter power ratio

K = system constants; e.g., ferite modulator, cable loss, etc.

$G_r G_t$ = antenna receive/transmit gain correction.

Several types of algorithms are available for performing the estimate defined in Equation 8.1, each differing generally in the way that the value of A, ground cell area, is defined. This system uses the Constant Angular Resolution ground cell as discussed in Section 4.3. As the name implies, this type of resolution provides for a ground cell in which the angle $\Delta\theta_{Di}$, Figure 8.1, remains constant with viewing angle; i.e., $\Delta\theta_{Di} = \Delta\theta_{Di+1}$. In this type of resolution cell the initial value of $\Delta\theta_{Di}$ would normally be selected such that the first cell length near nadir, L_1, Figure 8.1, is the minimum resolution required. Since L_i and θ_{Li} are related by

$$L_i = H\{\tan(\theta_{Di} + \frac{\Delta\theta_{Di}}{2}) - \tan(\theta_{Di} - \frac{\Delta\theta_{Di}}{2})\} \tag{8.2}$$

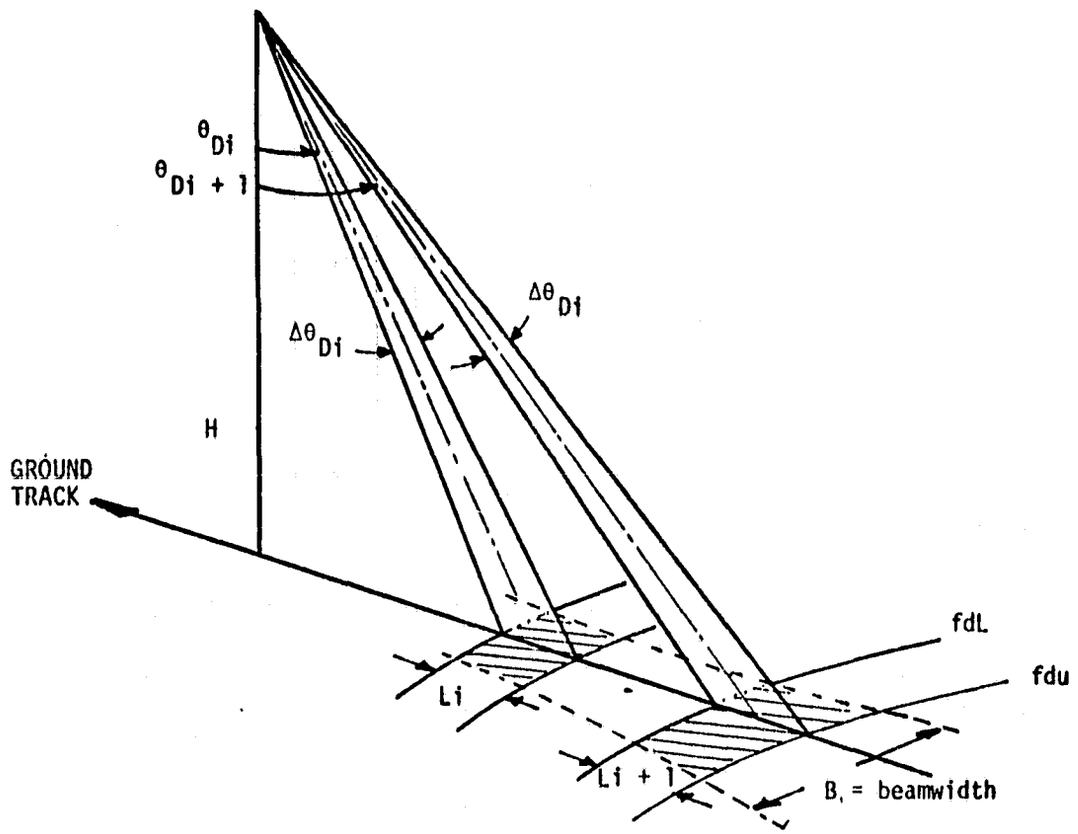


FIGURE 8.1 CONSTANT ANGULAR RESOLUTION CELLS

the L_i are specified for each θ_{Di} selected. The integration time would be the same for all θ_{Di}; specifically,

$$TI = \frac{L_1}{2V} \tag{8.3}$$

The desired bandwidth for angle θ_{Di} is given by

$$Bd_i = (2V\, L_i \cos^3\theta_{Di})/\lambda H \tag{8.4}$$

where V = aircraft ground speed vector,

L_i = instantaneous length of cell i, along ground track vector,

θ_{Di} = viewing angle, referenced to ground track vector,

λ = transmitter wave-length, and

H = aircraft altitude.

The desired doppler center frequency for angle θ_{Di} is given by

$$fdc_i = (2\ V \sin\theta_{Di})/\lambda \tag{8.5}$$

Note that θ_{Di} is the viewing angle referenced to the ground track line; i.e., the intersection of fdc_i with the X-axis in Figure 8.2. Thus to evaluate Bd_i and fdc_i the value of θ_{Di} must be first calculated. For a given look-angle, θ_{Di}, the values of X_{Li} and Y_{Li} may be derived using the following coordinate transformations:

$$\begin{bmatrix} X'' \\ Y'' \\ Z'' \end{bmatrix} = \begin{bmatrix} B \\ 0 \\ -A \end{bmatrix} \tag{8.6}$$

where $A = H/\cos\psi$, $B = X'' = X' = (r_L^2 - (Y')^2)^{\frac{1}{2}}$, $Y' = -H\tan\psi$ and $r_L = Z' \tan\theta_{Li}$.

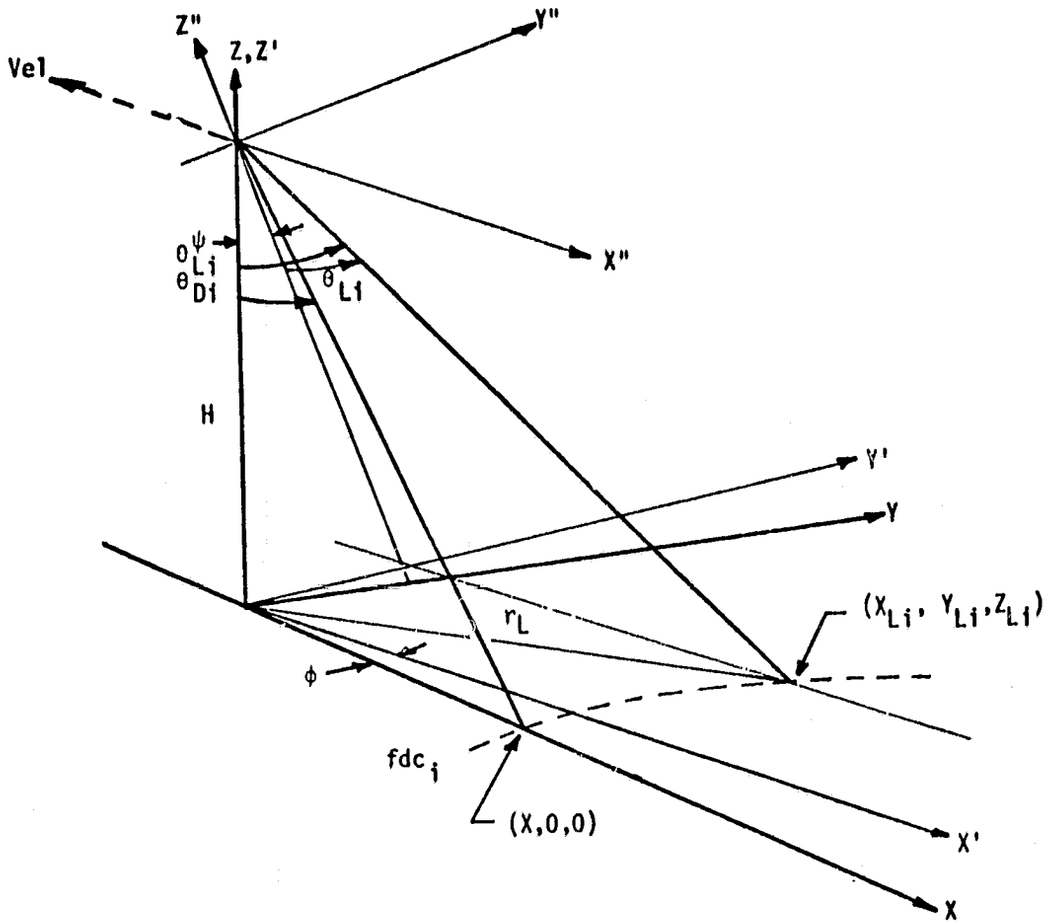


FIGURE 8.2 DOPPLER AND VIEWING ANGLE GEOMETRY

The two vectors

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} X'' \\ Y'' \\ Z'' \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ H \end{bmatrix} \tag{8.7a}$$

and

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \tag{8.7b}$$

give the expressions

$$X_{Li} = (-H/\cos\psi) \tan \theta'_{Li} \cos\phi + H \tan \psi \sin\phi \tag{8.8}$$

$$Y_{Li} = (-H/\cos\psi) \tan \theta'_{Li} \sin\phi - H \tan \psi \cos\phi \tag{8.9}$$

where $\tan \theta'_{Li} = -(\tan^2\theta_{Li} - \tan^2\psi)^{\frac{1}{2}}\cos\psi$, (8.10)

so that $\theta_{Di} = \mathrm{Tan}^{-1}(X_{Li}^2/(H^2 + Y_i^2))^{\frac{1}{2}}$. (8.11)

Notice in Equation 8.10 that when the roll angle exceeds the viewing angle, ($\psi > \theta_{Li}$), the expression becomes indeterminate. Therefore, care must be taken when computing $\tan \theta'_{Li}$, or the machine results will be unpredictable.

Because of this interaction between θ_{Li} and ψ, minimum values are imposed upon the values of θ_{Di} and θ_{Li}. The minimum Doppler angle, θ_{Dmin}, may be derived by considering that the minimum allowable doppler center frequency, fdc_{min}, is $BW_i/2$; i.e., half the bandwidth of the viewing angle nearest nadir. Equating the two expressions (for the

near nadir case)

$$BW_i/2 = \frac{L_1 V \cos^3\theta_{Dmin}}{\lambda H} \tag{8.11a}$$

and

$$fdc_{min} = \frac{2V\sin\theta_{Dmin}}{\lambda} \tag{8.11b}$$

gives the expression

$$\tan\theta_{Dmin} \geq \frac{L_1}{2H} \cos^2\theta_{Dmin} \tag{8.11c}$$

The solution for θ_{Dmin}, of course, depends on the parameter $\frac{L_1}{2H}$. For example, at H = 457.3 meters and L = 20 meters $\theta_{Dmin} \simeq 1.25°$.

Using rationale similar to the above, the minimum allowable viewing angle may also be derived. First, note in Figure 2 that geometric consideration indicate that whenever fdc $\geq$ BW/2 then X $\geq$ L/2. Using vector expressions for the coordinate system in Figure 8.2 the value of X is given by

$$X = X'\cos\phi - Y'\sin\phi \tag{8.11d}$$

Substituting for X' and Y' gives

$$\tan\theta'_L \geq \sin\psi \tan\phi - X \frac{\cos\psi}{H\cos\phi} , \tag{8.11e}$$

but at θ_{Lmin}, X = L/2 and, using Equation 8.10,

$$\tan^2\theta_{Lmin} = K^2 + \tan^2\psi \tag{8.11f}$$

where

$$K = -\tan\psi \tan\phi + \left(\frac{L}{2H}\right)\left(\frac{\sec\phi}{2}\right) \qquad (8.11g)$$

As in Equation (8.11c), the solution depends on the parameter L/2H.

Analysis of the function in Equation (8.11f) indicates that if roll and drift are bounded by $-10° \leq \psi \leq 10°$ and $-10° \leq \phi \leq 10°$ then the lower bound of $|\theta_L|$ is as shown in Figure 8.3. Since the value of the desired $|\theta_L|$ is normally $\geq 5°$ and $-5° \leq \psi \leq 5°$, the limitations do not represent a large penalty in viewing angles, however, the system software must account for the above limitations in order to avoid Doppler bands that extend off the end of the filter bank.

Note that the value of θ_D must also be properly limited. If the value of X_L is always $\geq L/2$, θ_D will automatically be correctly limited. These two limitations must be observed in the software if valid area and power ratios are to be calculated when using viewing angles such that $|\psi| \geq |\theta_L|$.

The procedure recommended and used herein is to compare the absolute value of the expected or desired look-angle, θ_{Li}, with the absolute value of roll, ψ, and whenever $|\psi| > |\theta_{Li}|$, set the value of $\theta_{Li} = -|\psi|$ for aft viewing angles, or generally, $\theta_{Li} = \frac{\theta_{Li}}{|\theta_{Li}|}|\psi|$. In addition, the value of X_L calculated in Equation 8.8 must be monitored and always set $\geq L/2$. Test cases which have been run at TAMU using this procedure indicate that satisfactory results are obtained for $|\psi| \leq 6°$, $|\phi| \leq 8°$, H = 457.3 meters, $10 \leq L \leq 50$ meters. Since the actual θ_{Li} is given in the output, no loss of information occurs.

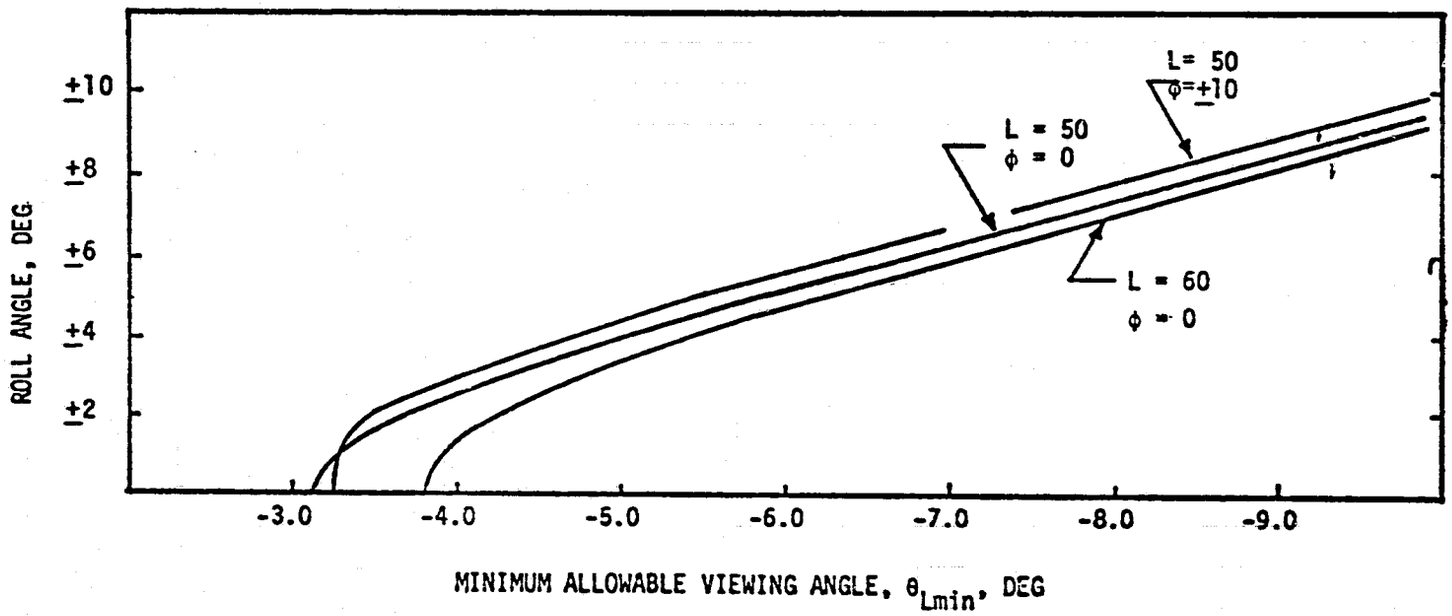


FIGURE 8.3 LIMITS ON VIEWING ANGLE, θL_i

Again, θ_{Di} must be used to evaluate the desired doppler center frequency, fdc_i, and to evaluate the desired bandwidth, Bd_i. However, when evaluating G_rG_t, the receiver/transmitter gain, the proper angle to use is θ'_{Li} given by Equation (8.10), but adjusted for actual Doppler center frequencies. This adjustment is needed because the actual Doppler center frequency is not always the expected center frequency. This shift in frequency occurs because a number of discrete spectral lines are used to make up the desired bandwidth, and the center of this set will not always correspond to the center of the desired bandwidth. For the same reason the desired and actual bandwidth are not the same, as illustrated in Figure 8.4.

The number of filter indexes required to represent the desired bandwidth is given by the expression

$$NF_i = [Bd_i/\Delta f + 0.5]_I \tag{8.13}$$

The actual bandwidth is given by

$$BA_i = NF_i\ \Delta f \tag{8.14}$$

The center frequency index may then be calculated (for NF_i odd) by

$$NC_i = [fcd_i/\Delta f + 0.5]_I \tag{8.14a}$$

or (for NF_i even) by

$$NC_i = [fdc_i/\Delta f]_I \tag{8.14b}$$

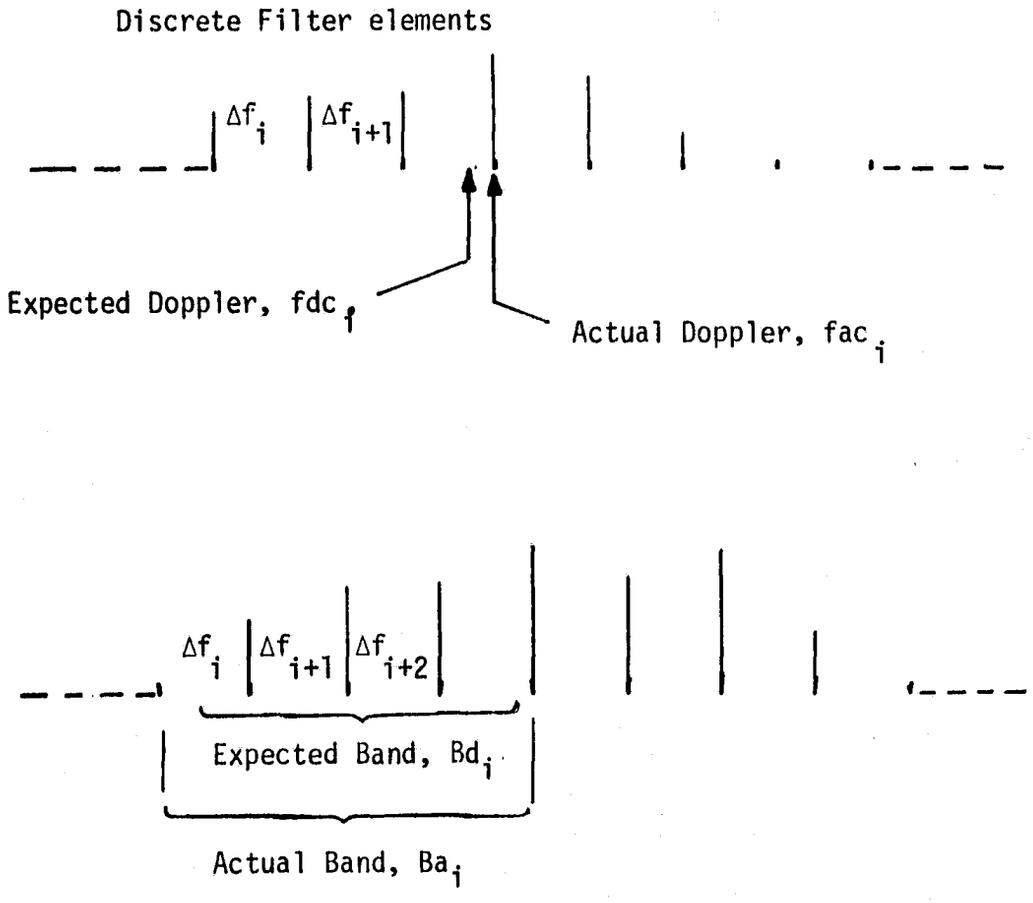


FIGURE 8.4 DESIRED AND ACTUAL BANDWIDTH DIFFERENCES

The left most index pointer for band BA_i is given (for NF_i odd) by

$$PRL_i = NC_i - (NF_i - 1)/2 \tag{8.14c}$$

or (for NF_i even) by

$$PRL_i = NC_i + 1 - NF_i/2 \tag{8.14d}$$

If NF_i is odd the actual center frequency is

$$fac_i = [fdc_i/\Delta f + 0.5]_I \, \Delta f \tag{8.15}$$

When the value of NF_i is even the actual center frequency is

$$fac_i = \{[fdc_i/\Delta f \;\;]_I + 0.5 \;\}\Delta f \tag{8.15a}$$

From Figure 8.2 it can be seen that the actual center frequency places the actual viewing point (X_{Li}, Y_{Li}, X_{Li}) on a different doppler line that expected, thus θ_{Li} and θ'_{Li} need to be updated before they can be used to evaluate exactly the cell area, range and gain. This may be done by first evaluating the new doppler angle along the ground track using actual center frequency, fac_i:

$$\theta_{TDi} = Sin^{-1} \, (fac_i \lambda/2V) \tag{8.16}$$

Next, the actual value of X_{Li} may be calculated by

$$X_{Li} = \{(H^2 + Y_{Li}{}^2) \tan^2 \theta_{TDi}\}^{\frac{1}{2}} \tag{8.17}$$

so that the true viewing angle is

$$\theta_{TLi} = \text{Tan}^{-1} \{ \frac{(XT^2_{Li} + Y_{Li}^2)^{\frac{1}{2}}}{H} \} \tag{8.18}$$

and true Range is

$$R_i = H/\cos\theta_{TLi} \tag{8.19}$$

The true value of θ'_{Li} for use in the gain tables is given by

$$\theta'_{TLi} = \text{Tan}^{-1}\{-\cos\phi \ (\tan^2\theta_{TLi} - \tan^2\psi)^{\frac{1}{2}}\} \tag{8.20}$$

The true viewing angle would be the appropriate value to be placed in the output data set with each estimated value of sigma-zero.

The calibration tone, similar to the individual Doppler center frequencies, does not lie in the center of the filter element in which it is found in the PSD (Power Spectral Data) filter bank; therefore, appropriate corrections to the measured power level in that filter must be made. The measured power transmitted that appears in the calibration-tone filter must be corrected by

$$PR = PR_M \{ \frac{(2N-1)^2}{[W(\Omega)]^2} \} \tag{8.21}$$

where N = 512, number of filter elements,

$\Omega = 2\pi f_c/f_s - M_c \frac{2\pi}{N}$, where

$M_c = [(f_c/f_s)512 + 0.5]_I + 1$,

f_s = sample frequency,

f_c = calibration tone frequency, and

PR_M = the measured power in the calibration tone filter.

and

$$W(\Omega) = \sin\{(2N-1)\ \Omega/2)/\sin\ (\Omega/2)$$

In the above discussion equations were presented which provide the exact solution to the radar equation for sigma-zero estimates. In an airborne, near real-time processor such as this implementation it is appropriate to consider more efficient and expeditious means of calculating these solutions. The only term of Equation (8.1) that promises any significant savings in computations is (R^4/A).

Looking first at Range, it can be shown that computing Range, R, by the expression

$$R_i = H/\cos\ \theta_{Li} \tag{8.22}$$

as opposed to Equation (8.19), which used θ_{TLi}, the true viewing-angle, will result in a Range errors of up to $\pm$ 2.0% for $|\theta_{Li}| \leq 45°$, with $-6° \leq \psi \leq 6°$ and $-8° \leq \phi \leq 8°$. These errors in the combined term (R^4/A) can produce sigma-zero values off by as much as $\pm$ 0.45db.

Two alternative approximation equations for area were considered:

$$A = LW/\cos\phi\ \cos\psi \tag{8.22a}$$

and

$$A = W'(Y2-Y1)/\cos\phi \tag{8.22b}$$

Equation (8.22a) is a simple rectangular approximation where

$$L = H\{\tan\theta_2 - \tan\theta_1\}/\cos\phi \qquad (8.22c)$$

θ_2 = Doppler angle for fdu, upper frequency

θ_1 = Doppler angle for fdL, lower frequency

and

$$W = 2R \tan(B/2)/\cos(\psi) \qquad (8.22d)$$

$\bar{B}$ = Beamwidth referenced to the origin of the aircraft coordinate system

Equation (8.22b), whose terms are defined in Reference [13], is a more rigorous approximation to the actual ground cell shape. Although computation time is somewhat longer for Equation (8.22b), it is considered to be the best alternative because it offers up to 10% more accuracy at viewing angle where $|\theta_{Li}| \geq 45°$ and with roll and drift excursions of up to $\pm$ 6° and $\pm$ 8° respectively.

Recalling from Section 8.1.1.1, Equation (8.2) must be solved eight times for eight values of cell length. By using small angle approximations an equivalent expression is

$$L_i = \frac{H\Delta\theta}{\cos^2\theta_{Li}\cos^2\frac{\Delta\theta}{2}} \qquad (8.23)$$

which is accurate to $\pm$ 2.0% for $\theta_{Li} \leq 45°$ and $|\theta_{Li}| \geq |\psi|$. For ground cells defined by Constant Angular Resolution the value of $\Delta\theta$, and consequently $\cos^2\frac{\Delta\theta}{2}$, are constant; therefore, the cell lengths would involve only the calculations

$$Li = \frac{H\ \Delta\theta'}{\cos^2\theta_{Li}} \tag{8.24}$$

where $\Delta\theta' = \Delta\theta/\cos^2\Delta\theta/2$, and is a stored constant.

For viewing angles several times grater than the aircraft roll angle (ψ) other approximations can be made; e.g., $\theta_{Di} \simeq \theta_{Li}$ and for $\psi \simeq 0.0$, $\theta'_{Li} \simeq \theta_{Li}$ in the Gain tables. However, scatterometer data processing done to date indicates that $\psi \simeq 0.0$ is not generally valid, therefore, the software algorithm presented in this design will account for aircraft roll angles.

8.1.2 System Constants

Equation (8.1) may be re-written in the form

$$\hat{\sigma}^o_i = \frac{(4\pi)^3}{\lambda^2} \cdot \frac{C_L}{K} \cdot \frac{Z_w(fdc_i)}{GG(\theta'_{Li})} \cdot \frac{R^4(\theta_{Li})}{A(\theta_{Li})} \cdot \frac{(P_R(\theta_{Di})}{P_T}$$

Where C_L = cable loss

K = calibration constant, plus other corrections as needed

$Z_w(fdc_i)$ = water roll-off filter correction for center frequency fdc_i

$GG(\theta'_{Li})$ = antenna gain at antenna viewing angle θ'_{Li}

$P_R(\theta_{Di})$ = receiver power at doppler angle θ_{Di}

and

P_T = transmitter power.

The system constants will include the first three terms. The first two constants, $(4\pi)^3/\lambda^2$ and C_L, remain fixed for a selected frequency. It is assumed that K may vary between horizontal, vertical, like and cross polarization combinations, therefore, four values should be reserved in

the software; i.e., K_{VV}, K_{VH}, K_{HH}, K_{HV}. Similarly, there must be four segments in the gain tables, four segments of beamwidth and a roll-off table for a processor that will handle all polarization combinations. Two degree increments with linear interpolation in these tables has been adequate in prior usages. Normally, antenna data are not sufficiently accurate to warrant more refined tables. When indexing into the constant tables the software will use two flags, one indicating polarization of the transmitter and the other showing the polarization of the receiver. These flags will be combined to compute the address of the proper segment in the table. To make the address computation most efficient the tables should be ordered the same way for all constants; e.g., Figure 8.5, gain values are ordered as GG_{VV}, GG_{VH}, GG_{HV}, and GG_{HH}. By setting the two polarization flags as indicated below it will be easy to compute the correct segment address into each table knowing only each starting address.

Transmit/Receive Polarization Flag	Polarization Combination
00	VV
01	VH
10	HV
11	HH

If constant beamwidth values are used, as is currently being done with L-band data processing, no beamwidth tables will be required. Also, it is possible to use a function to compute roll-off, rather than a table, however, table look-up will be faster than function-evaluation, therefore a table for roll-off should be included.

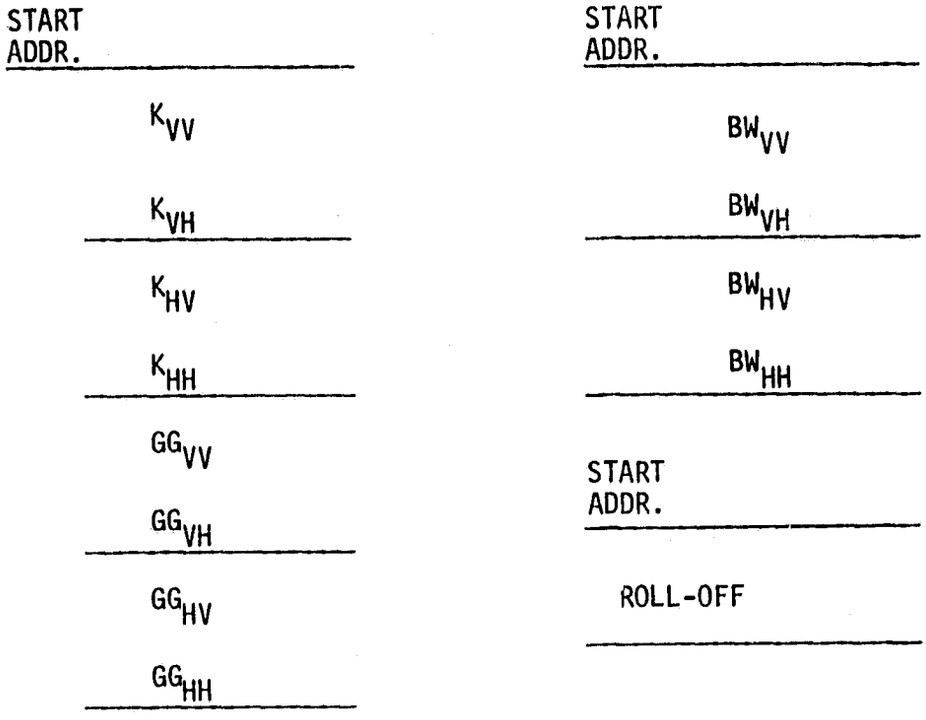


FIGURE 8.5 CONSTANT TABLE ARRANGEMENT

There are a number of other constants, such as $\Delta\theta'$, T_s, f_s, etc., associated with individual computation modules. These will be stored in the same region of core as the module itself.

8.2 Software Architecture

8.2.1 General Approach

The software system design features a modular approach with the structure boundaries defined by specific functions that are performed. Simplicity of operation and minimum operator interaction have been the governing considerations in laying out the system.

In general the system is defined by three specific "states" of operation, Fig. 8.6. These three states form the highest levels of the software architecture. The purpose of each operating state is to perform a specific task when directed by the system operator through entries on the system key board.

The "RESET" state not only performs all initializations in preparation for the "RUN" state, but also performs all communications between the system and the operator.

The "RUN" state performs all sigma-zero calculations using data acquired from the NERDAS frame and the radar signal integrator. The calculated data is aligned and formatted, along with appropriate NERDAS data, flags and alarms, and sent to the data output.

The "HALT" state is simply a standby state for the processor. Once it enters this state and completes its close out functions, the system idles until a "RESET" command is received from the operator through the keyboard.

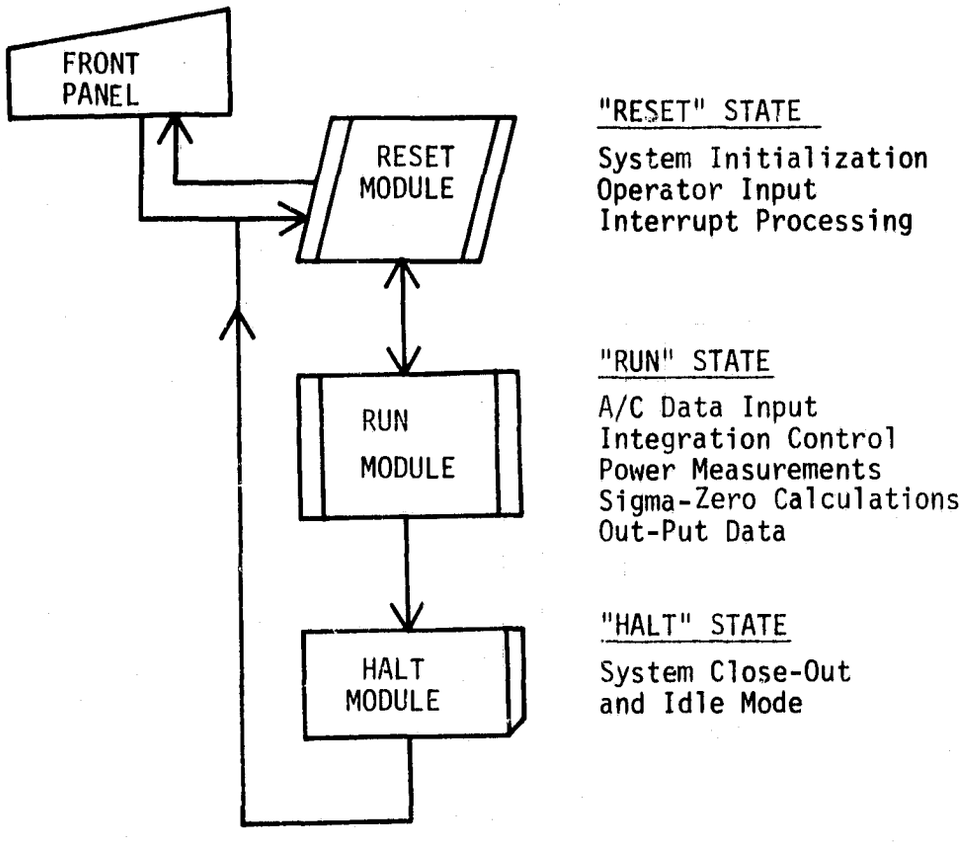


FIGURE 8.6 C-BAND SCATTEROMETER SOFTWARE SYSTEM CONCEPT

8.2.2 Operator Interface

All operator commands are entered through the keyboard and are received and processed by the "RESET" module as program interrupts. The software system Interrupt Processor, a part of the RESET module, is designed to respond only to those interrupts which are appropriate to the current "state" in which the system is operating. For example, while in the RESET state, the integration timer (Interrupt Nbr. 1) is ignored, all others shown in Figure 8.7 are processed.

The keyboard may be viewed as a set of three types of keys; i.e.,

Key Type		Key Name
State	:	Reset, Run, Halt
Function	:	Enter, Display, Clear
Digit	:	Values 0, 1,...,9. '.', CHS (change sign) and Vel, Alt, Drift, Roll, Pitch

Any time a DIGIT key is pressed by the operator it creates an Interrupt-0, Figure 8.7. All such interrupts are processed by the RDKEY routine, a part of the RESET module. This type of interrupt will be processed only while in the RESET state, therefore, all aircraft parameter over-ride values must be entered while in this state. FUNCTION keys also create interrupts which are processed only during operation in the RESET state. These are Interrupts-4, -5, and -7 which are used in conjunction with the DIGIT keys to enter values of aircraft over-ride parameters and to display or clear specific values already entered.

The operator procedure for entering over-ride values must have

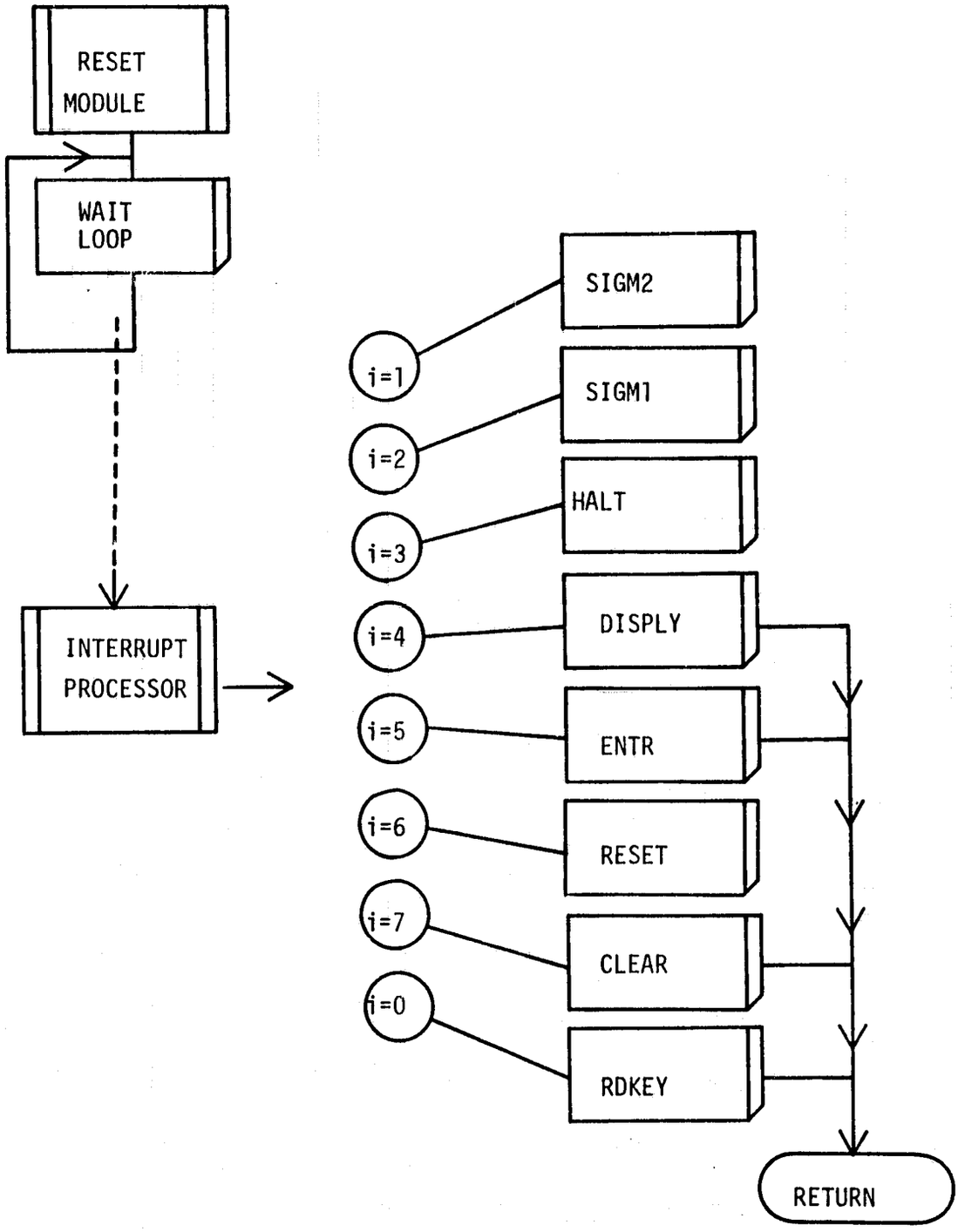


FIGURE 8.7 VECTORED INTERRUPT CONTROL STRUCTURE

three properly sequenced phases; i.e., value, command, destination. For example, to over-ride the NERDAS values of altitude, the following sequence would be used:

a. key in the string '1500', (value)

b. key 'ENTER' (command)

c. key 'ALT' (destination)

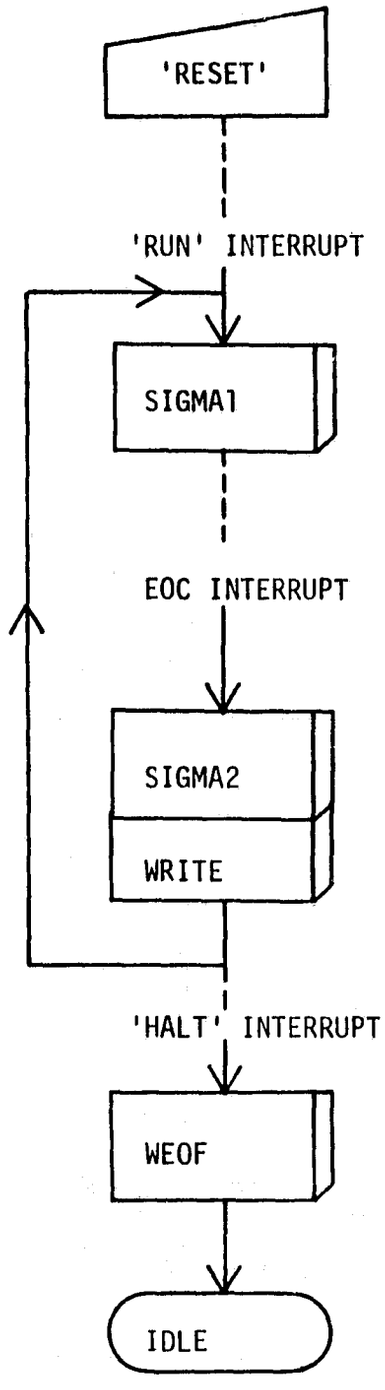
This sequence results in the over-ride value of 1500 feet being used in all sigma-0 calculations rather than the NERDAS values.

Depressing STATE keys such as 'RUN', 'HALT', or 'RESET' cause the software to process Interrupt-2, -3, or -6, respectively. As illustrated in Figure 8.7, the Interrupt-2 causes the processor to begin calculating sigma-zero values. The system will remain in this state as long as power is supplied, and will process only Interrupt-1 or Interrupt-2 (the HALT command). Thus any changes to the over-ride structure must be made by bringing the system first to HALT then RESET if already operating in the RUN state.

Prior to putting the system into the "RUN" state the operator should select the appropriate position on the Receiver Polarization switch.

8.2.3 Continuous Product Concept

Once the system is placed in the 'RUN' state, values of sigma-zero, based upon input NERDAS data, are calculated continuously for each of a set of eight angles. As each set of eight sigma-zero values is calculated, a set is transferred to the Bi-ϕ L output port for tape recording. As long as valid NERDAS and radar data are received, sigma-



SIGMA1:
RESET INTERRUPTS (HALT & EOC)
READ NERDAS, SET ALARMS AND
OVER-RIDE VALUES
SET AND START INTEGRATORS
CALCULATE:

$$\sigma^{o'}_i = \frac{(4\pi)^3}{\lambda^2} \frac{R_i 4}{K_s} \frac{Z_{wi}}{A_i}, \quad i = 1, 8$$

SIGMA2:
READ NERDAS DATA,
READ FILTERS, F_i, $i = 1, 12$
COMPUTE:

$$\hat{\sigma}^{o}_i = \sigma^{o'}_i \, PR_{Li}/GG_{Li}, \text{ or}$$

$$\hat{\sigma}^{o}_i = \sigma^{o'}_i \, PR_{Xi}/GG_{Xi}, \quad i = 1,8$$

FINISH LOADING BUFFER LINE
WRITE:
OUTPUT A LINE TO Bi-ϕ L

FIGURE 8.8 4.75 GHz SCATTEROMETER SYSTEM SOFTWARE

zero values will be produced at the data output port -- thus providing a continuous product for the user.

The cycle time, or time delay between each record at the output, will be a function of the microprocessor overhead time and the integration time. The integration time is radar frequency and aircraft ground velocity related and thus is somewhat independent of any software optimization scheme. However, the microprocessor duty cycle can be adjusted within certain limits to minimize the overhead. This should be done during program development testing by properly balancing tasks between the two sigma-zero calculation sections, SIGMA1 and SIGMA2, of the RUN module, Figure 8.8.

8.2.4 "RESET" Module Algorithm

This module provides all program Interrupt Service, as illustrated in Figure 8.7. From the operator's point of view, it is the module through which all communications are made with the software. In addition to operator inputs, the module handles all "pre-RUN" housekeeping functions such as Interrupt Structure set-up, Input/Output port initializations, Data Buffer initializations and resets all flags, alarms and the Front Panel.

The RESET algorithm can be stated as below where the notation $^{C}(ABC) \leftarrow 0$ means that the value zero is assigned to the location ABC:

1. Disable Interrupts
2. Initialize the Interrupt Controller (8259) to accept all but Interrupt-1, initialize the CZT-Board with DMA address, and set-up the Bi-ϕ L board.
3. Reset Data Buffers, and their Flags, and Pointers

4. Reset Front Panel

5. Reset System Alarms and Flags

6. Reset Interrupt Processor
   a. ^{C}(STACK) ← 0
   b. ^{C}(DECFG) ← 0
   c. ^{C}(DISFLG) ← 0
   d. ^{C}(ENTER) ← 0
   e. ^{C}(X) ← 0
   f. ^{C}(NNBR) ← 1

7. Enable Interrupts

8. Wait for Interrupt i, i = 0, 2, 3, ..., 7

   a. If Received, transfer control to the appropriate processor, otherwise go to 8.

The above algorithm is illustrated in flow chart form in Appendix F, Chart 1.0. Each step is represented by a box in the chart. Each box has a number whose second digit corresponds to the above step numbers.

While operating in this module the Interrupt-1 (i=1), Figure 8.7, will not be accepted, i.e., it is masked out during the set-up, Step 2, above. Although the Interrupt-3 is acceptable, no purpose is served by invoking it in this mode of the operation of the system.

The Interrupts-0, -2, -3,..., -7 are each created by the operator when a particular key on the keyboard is depressed. Table 8.1 summarizes the key types, names, and their respective assignments within the structure.

TABLE 8.1 INTERRUPT/KEY ASSIGNMENTS

KEY TYPE	KEY NAME (LABEL)	INTERRUPT CREATED
DIGIT	0, 1, 2,...,9, ·, CHS, VEL, ALT, DRFT, ROLL, PITCH	INTERRUPT-0
STATE	RUN	INTERRUPT-2
	HALT	INTERRUPT-3
	RESET	INTERRUPT-6
FUNCTION	DISPLAY	INTERRUPT-4
	ENTER	INTERRUPT-5
	CLEAR	INTERRUPT-7

Whenever the operator, while in the RESET mode of operation, depresses a key, program control is transferred directly to one of the seven interrupt processing modules, Figure 8.7, tagged with i = 0, 2, 3,...,7. Each of these modules has its own special purpose algorithm. Starting with i = 0, each will be outlined in enough detail to illustrate its principal functions and create an understanding of the Flow Charts in Appendix F.

The RDKEY module handles all digit key activities. These activities are all related to entering the appropriate aircraft data over-ride values. Again, the algorithm is activated when a digit key, Table 8.1, is depressed. The RDKEY algorithm may be stated as follows (the notation ^{C}(ABC) means the content of Location ABC):

1. Read key value into a "save" location, INDIG.
2. Interpret its value and
   a. If the value $\geq 11_{10}$ go to DECIM, Step 7.
   b. If the value $\geq 10_{10}$ go to CHS, Step 9.
   c. If the 'ENTER' key has been depressed go to PARM, Step 11.
   d. If this is the 6th digit go to ERR1, Step 14.
3. Save digit character and echo character to the Front Panel.
4. Add 1 to the digit counter.
5. Send End-of-Interrupt (EOI)
6. Return to WAIT LOOP.

---

7. If DECFG $\neq$ 0 go to RETURN, Step 5.
8. Set DECFG, go to Step 3.

9. Multiply ^{C}(NNBR) by -1.
   a. If ^{C}(NNBR) > 0 ECHO '+' to Front Panel.
   b. If ^{C}(NNBR) < 0 ECHO '-' to Front Panel.

10. Go to Return, Step 5.

---

11. If the 'ENTER' key has not been depressed go to DISPL, Step 15.

12. If the digit, ^{C}(INDIG), is $\leq$ 3 go ERR1, Step 14.

13. If ^{C}(INDIG) $\leq$ 8 go to ACPRM, Step 20.

14. ERR1, (entry procedure error),
   a. Disable all interrupts except 'CLEAR' or 'RESET'.
   b. Flash Front Panel at 500 msec intervals.

---

15. If ^{C}(DISFLG) = 0 go to ERR1, Step 14.

16. If ^{C}(INDIG) > 9 go to ERR1, Step 14.

17. Display parameter name and value on the Front Panel.

18. Reset stack and flags:
   a. ^{C}(STACK) $\leftarrow$ 0
   b. ^{C}(ENTER) $\leftarrow$ 0
   c. ^{C}(DECFG) $\leftarrow$ 0
   d. ^{C}(DISFLG) $\leftarrow$ 0
   e. ^{C}(X) $\leftarrow$ 0
   f. ^{C}(NNBR) $\leftarrow$ 1

19. Go Step 5.

---

20. Save parameter string and sign

21. Save parameter value

22. Set bit in Over-ride Flag word

23. Display '*' in appropriate LED on the Front Panel.

24. Go to Step 18.

The above algorithm is illustrated in flow chart form in Appendix F, charts labeled Interrupt-0.

Moving up the stack of modules in Figure 8.7, the CLEAR, Interrupt-7, processor has a relatively simple algorithm:

1. Blank the digit LED's on the Front Panel.
2. Reset the Interrupt Processor (Go to Step 6 of the RESET module).

Next up the stack is the Interrupt-6, created by pressing the RESET key. This function does the same thing from the software point of view as turning power off and on again. It simply transfers program control back to Step 1 of the RESET algorithm.

Interrupt-5, handled by the ENTR module is also relatively simple. Its function is to signal the software that the next key depressed will designate the destination of the parameter already keyed into the panel. It may be stated as follows:

1. ^{C}(ENTER) ← 1
2. Flash Front Panel LED's one time to signal acceptance.
3. Go to Step 5 of RDKEY module, (Return to Wait Loop).

The Interrupt-4 function interacts with the RDKEY module. By depressing the DISPL key the software is signaled that the next digit key pressed will designate the contents to be displayed. The algorithm follows:

1. ^{C}(DISFLG) ← 1
2. ^{C}(ENTER) ← 2
3. Go to Step 2 of ENTR module, (flashes Front Panel and returns to Wait Loop).

Now, the operator, by pressing the appropriate digit key next will cause the control to pass through the RDKEY module to Step 18.

The HALT module processes Interrupt-3, (i = 3, Figure 8.7). Its function is to perform an orderly close out of the data production operation, SIGMA1 and SIGMA2 modules. It does the following:

1. Disable Interrupts and reset mask for Interrupt-6 only.
2. Build a 'Fill' frame for End-of-File identification.
3. Write 50 'Fill' frames into the output buffer.
4. Reset Front Panel
5. Enables Interrupts
6. Wait for Interrupt-6.

The remaining two Interrupts in Figure 8.7 are associated with the 'RUN' mode of operation. When the function key 'RUN' is depressed, control transfers directly to the SIGMA1 module, Figure 8.8. While in this mode of operation the integration time on the CZT-board, will provide the Interrupt-1, (i = 1, each time it "times-out", or completes the cycle of integration provided to it by the SIGMA1 module.

8.2.5 "RUN" Module Algorithm

As discussed in Section 8.1, the objective of the software is to repeatedly solve the estimated scattering coefficient equation

$$\hat{\sigma}^{o}_{i} = \frac{(4\pi)^3}{\lambda^2} \frac{R_i 4}{A_i} \frac{Zw_i}{K_s} \frac{PR_i}{GG_i} \tag{8.25}$$

This is done for each of eight pre-selected viewing angles with either like or cross polarized input data. These viewing angles, θ_{Li}, i=i, 8,

are pre-loaded into the software object code and are not changed except by reprogramming the entire software package.

For each operating cycle (an output data set) the software will analyze the input data calibration tones to determine if horizontal (H-) or vertical (V-) polarized data is being transmitted. A front panel input will be sensed by the software to determine whether the receiver is in Horizontal or Vertical polarization mode.

Having resolved these variables, the software system will calculate in sets of eight one of two forms of Equation (8.1); i.e.,

Like-Polarized:

$$\hat{\sigma}^{o}_{Li} = \sigma^{o'}_{i} \; PR_i / GG_{Li} A_{Li} \tag{8.26}$$

Cross-Polarized:

$$\hat{\sigma}^{o}_{xi} = \sigma^{o'}_{i} \; PR_i / GG_{xi} A_{xi} \tag{8.26a}$$

where the parameters are

PR_i, received/transmit power ratio in band i;

$\sigma^{o'}_{i}$, $\dfrac{(4\pi)^3}{\lambda^2} \; \dfrac{R_i^4 Z_w(f_i)}{K_s}$ for band i;

GG_{xi}, Antenna gain for viewing angle i, cross-polarized gain table.

A_{Li}, ground cell area for viewing i using like-polarized beamwidth tables.

A_{xi}, ground cell area for viewing angle i using cross-polarized beamwidth tables.

When indexing into the beamwidth and gain tables the software will use the polarization tone and front panel input to determine whether to use the horizontal or vertical tables and whether to use the Like or Cross section in the table. Note that if a constant beamwidth value is used for all polarization combinations, the computation of A_i will be done as part of the $\sigma^{o}{}'_i$ parameter.

The system of calculation defined for $\hat{\sigma}^{o}$ estimation in modules SIGMA1 and SIGMA2 presume that the hardware is structured for single channel CZT operation, Figure 8.9; however, with appropriate changes in the set-up, control, and output portions of the SIGMA1 and SIGMA2 algorithms the software package could be used with a two-channel hardware system, Figure 8.10. The implications of using a system like that in Figure 8.10 are: 1) the processor must handle twice the data, either serially or parallel, thus the ground cell to ground cell separation will increase significantly, 2) the memory buffer size needed doubles for both the CZT and the microprocessor (μP), 3) the input/output times are doubled, and 4) the software must have a means of knowing which channel has the Like-Polarized data and which has the Cross-Polarized, (this could be handled by always assigning Cross-Polarized data to CZT #2 as suggested by Figure 8.10).

8.2.5.1 SIGMA1 Sub-Module Algorithm, INTERRUPT-2 PROCESSING

In the most basic form the algorithm for the SIGMA1 module which calculates the σ'^{o} estimations may be stated as follows:

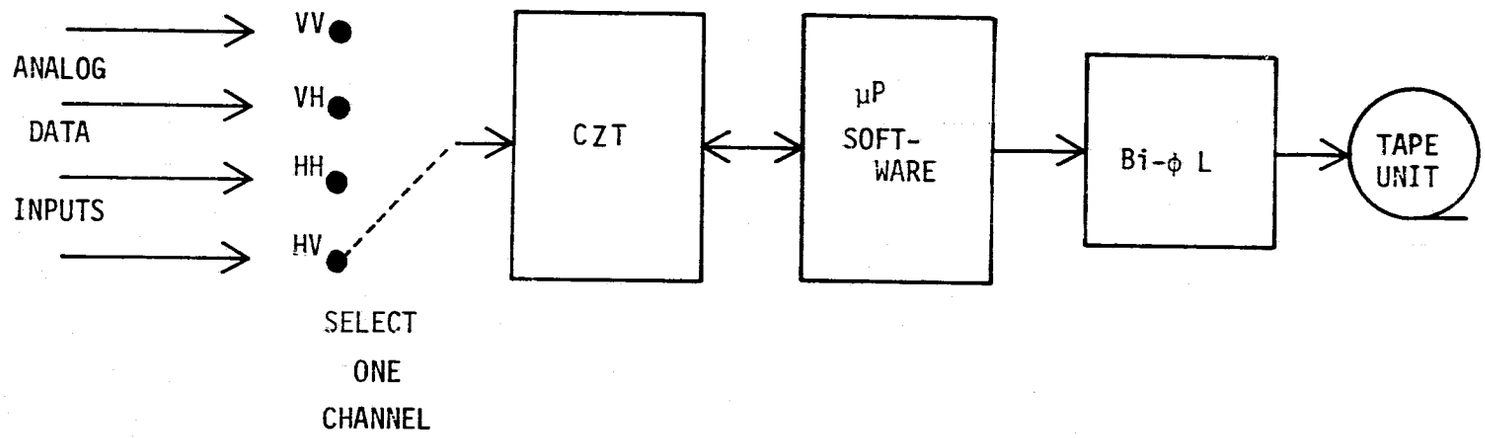


FIGURE 8.9. SİNGLE CHANNEL CZT SYSTEM

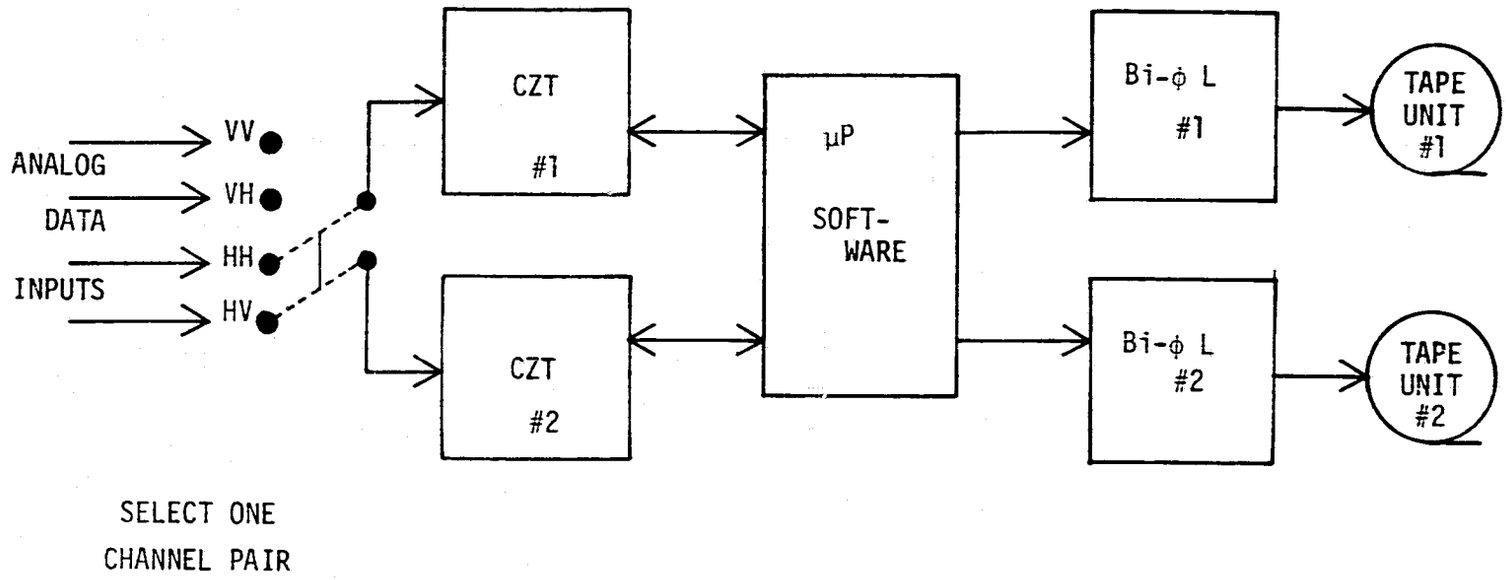


FIGURE 8.10. TWO CHANNEL CZT SYSTEM

SIGMA1:

1.1 Reset Interrupt mask to acknowledge the EOC and 'HALT' interrupts only. Read aircraft NERDAS and establish velocity, V; altitude, H; pitch, p; roll, ψ; drift, ϕ; and clock time, T_o.

1.2 Evaluate L_i, i=1, 8 using Equation 8.24.

1.3 Compute Integration time using L_1 from Step 2, Equation 8.3; start Integration on the CZT-board.

1.4 Compute Output Buffer length, NL.

1.5 Set values of load pointers and write pointers, up-date σ^o line pointers in the buffer. Move A/C data into buffer.

1.6 For i=1, 8 using θ_{Li}, do:

a. Compute θ'_{Li}, X_i, Y_i, θ_{Di}

b. Evaluate fdc_i, using Equation 8.5

c. Evaluate Bd_i, Equation 8.4

d. Compute NF_i, Equation 8.13

e. Calculate Ba_i, Equation 8.14

f. Evaluate fac_i, θ_{TDi}, θ_{TLi}, and R_i; Equations 8.15 through 8.19.

g. Using Ba_i and NF_i calculate index pointer, PRL_i (left most index), into the Power Spectral Density (PSD) data set, Equations 8.14a through 8.14d.

h. Calibration and Polarization index pointers are stored constants

i. Look up roll-off value, $Z(fac_i)$

j. If beamwidth is constant calculate area A_i.

k. Evaluate the term $\sigma^o{}'_i$, Figure 8.8.

1.7 Wait for Interrupt-1, (i=1, Figure 8.5).

8.2.5.2 SIGMA2 Sub-Module Algorithm, INTERRUPT-1 PROCESSING

Upon receipt of Interrupt-1 program control is passed directly to the SIGMA2 module which does the following:

SIGMA2 (AND WRITE):

1.1 Transfer CZT data to μP memory, after interrupt Flag checks and resets

1.2 Using pointers from Step 6.h, evaluate calibration and polarization power. Set polarization Flag Bits and calibration alarm (if appropriate).

1.3 For i=1, 8 do:

a. Evaluate PR_i, power received in band i

b. Look up G_rG_t, gain for antenna, (and if appropriate BMW_i, beam-width for θ_{Li}).

c. Compute A_i, area for ground cell i, unless already done in Step 6.k of SIGMA1.

d. Evaluate $\hat{\sigma}^o_{Li}$ or $\hat{\sigma}^o_{xi}$, equation 8.26a or 8.26b.

1.4 Store viewing angle values, $\hat{\sigma}^o_{Li}$ or $\hat{\sigma}^o_{xi}$ values, Alarms and Flags in the output buffer, formatted as shown in the Flow Charts.

1.5 Pack and write the appropriate output buffer line to the Bi-ϕ L interface.

1.6 Set the Pass Number, ^{c}(NPASS) $\leftarrow$ 1

1.7 Read aircraft NERDAS and establish V,H,P,ψ,ϕ, and T_k, the current clock time.

1.8 Using previous time, T_o, (SIGMA1, Step 1) compute last cycle time, T_c.

1.9 Move T_k to T_o.

1.10 Go to SIGMA1, Step 2, after clearing the Time Alarm.

The system will remain in the computational cycle until the operator depresses the 'HALT' key.

8.2.5.3 Additional Considerations for SIGMA1 and SIGMA2

In conjunction with preparing the software code which will execute the SIGMA1 and SIGMA2 algorithms there are additional considerations and details related to some of the above steps. These are summarized as follows:

SIGMA1: (Additional considerations):

Step 1 - The code for this step must include logic for a) selecting over-ride values in place of NERDAS data received, b) evaluating the values of V,H,P,ψ, and ϕ against some standard (if not over-ridden) and setting the computational value to the input or to the standard, c) setting a bit in the output data frame to indicate the type data used, and d) writing an '*' to the Front Panel LED when failure occurs. NERDAS failure must be defined prior to coding this logic. Buffering data into Output Data Line cannot be done here. Where appropriate, over-ride or default values should be substituted into the line.

Step 2 - In calculating L_i, i=1, 8, the expression in Equation 8.24, $L_i = (h\Delta\theta')/\cos^2\theta_{Li}$, the parameter $\Delta\theta'$ is a stored constant whose value is calculated by $\Delta\theta/\cos^2(\frac{\Delta\theta}{2})$ where $\Delta\theta = 3°$ (0.05236 radians), so that $\Delta\theta' = 0.05239$ radians.

Step 3 - Integration time for the kth cycle is calculated using Equation 8.3 where L_1 is the length of the ground cell associated with viewing angle θ_{L1}; i.e., the smallest θ_{Li}. Integration is started by sending the CZT Board the number of sub-records desired, NS, where

$$NS = [(T_I/512)f_s + 0.5]_I \ ,$$

or $$NS = [T_I/T_s + 0.5]_I \qquad (8.27)$$

where $T_s = 512/f_s$, and as before f_s = sample frequency. T_s and f_s will be stored constants.

Step 4 - Output buffer length is computed by

$$NL = [(H \tan\theta_{L8})/VT_c + 0.5]_I \qquad (8.28)$$

where θ_{L8} is the largest viewing angle and T_c is the cycle time. When ^{c}(NPASS) = 0, the first pass through SIGMA1, T_c will not have been evaluated, therefore the software should use a dummy value, $T_c = 2T_I$. The value of the tan θ_{L8} should be a stored constant. If the value of NL is greater than the reserved buffer length (200 lines) the software must set an alarm; i.e., byte 79H (xx H means Hexadecimal notation) in the output data frame, and the Buffer-Over-Flow LED on the Front Panel gets and '*' written into it. Finally, provisions must be made to set NL to the maximum buffer length value when the Over-flow occurs.

Step 5 - On the first pass through SIGMA1, ^{c}(NPASS) = 0, the buffer bias value is set to NL + 10; i.e.,

$$MX_o = NL_o + 10 \qquad (8.29)$$

represents the line number at which the output buffer loading begins. On each successive pass the value of MX is incremented so that in general

$$MX_k = MX_o + k \qquad (8.30)$$

where k = 0,1,2,..., 255 and NL_o is the initial value of NL. The value of k is simply the pass number. Care must be taken when the register accumulating MX over-flows, else a frame of data will be lost. This is averted by simply setting MX = 1 wherever over flow occurs. The input or load-line pointer is calculated by

$$LL_k = [MX_k/Mo]^R \tag{8.31}$$

where $[\ ]^R$ signifies the remainder of the integer division, and Mo is the modulus for the buffer. Its value is

$$Mo = NL_o + 20 \tag{8.32}$$

so that the value of LL_k cycles the load point through the buffer. Line pointers, LP_i, for saving each of the sigma-zero values and their associated viewing angle is evaluated by

$$LP_i = [[MX - NL(\frac{\tan\theta_{Di}}{\tan\theta_{L8}})]_I/Mo]^R \tag{8.33}$$

If $LP_i < 0$, set $LP_i = NL + LP_i$.

Column pointers into the buffer for each value will be determined by their respective byte position assigned in the Bi-ϕ L output line, see Flow Charts for the 'WRITE' module.
The index of the output line which is sent to the Bi-ϕ L interface is calculated by

$$KO_k = [(MX_k - NL_o)/Mo]^R \tag{8.34}$$

If KO < 0, set $KO = KO + NL_o$.

Step 6 - This step must first take care of those roll angles, ψ, which exceed the desired viewing angle θ_{Li}. Thus, the appropriate order of computation is as follows:

a. 1) if $|\psi| > |\theta_{Li}|$, $\theta_{Li} = -|\psi|$

2) $\tan \theta'_{Li} = -\cos\psi \, (\tan^2\theta_{Li} - \tan^2\psi)^{\frac{1}{2}}$

3) $XT = (-H/\cos\psi) \tan\theta'_{Li}$

4) $YT = H \tan\psi$

5) $X_i = XT \cos\phi + YT \sin\phi$, but if $X_i < L_i/2$, set $X_i = L_i/2$

6) $Y_i = XT \cos\phi - YT \cos\phi$

7) $XD = H^2 + Y_i^2$

8) $\theta_{Di} = \text{Tan}^{-1} \, (X_i^2/XD)^{\frac{1}{2}}$

b. Evaluate 6.b through 6.g as indicated.

c. Calibration and polarization index pointers are evaluated before program loading and are stored constants. Consideration must be given, however, to the type of scatterometer when these constants are calculated.
For C-band scatterometers the left most pointers into the PSD (Power Spectral Data) for calibration tone power are found for horizontal and vertical polarization by

$$PCH_L = \left[2860 \left(\frac{512}{f_s}\right) + 0.5\right]_I \tag{8.35}$$

and

$$PCV_L = \left[3280 \left(\frac{512}{f_s}\right) + 0.5\right]_I \tag{8.36}$$

when the tones are inserted on the lower sideband. However, if the insertion is on the upper sideband then

$$PCH_u = 512 - PCH_L \tag{8.37}$$

and

$$PCV_u = 512 - PCV_L \tag{8.38}$$

Note that all pointer expressions assume that the filter spectral lines have been ordered with respect to zero frequency; i.e., $\omega_0 = 0$, $\omega_1 = f_s/512$, $\omega_2 = 2f_s/512, \ldots,$ $\omega_i = \frac{i512}{f_s}$ for $i \leq 255$. However, zero-frequency (dc) actually appears in the CZT spectrum at ω_2, thus all pointers must be biased by +2. Also, this assumes that the discrete Fourier transform of X + jY has been implemented by the CZT board where X is the true cosine channel and Y is the true sine channel. If significant power is present at either PCH or PCV with respect to the noise level then either horizontal polarization or vertical polarization, respectively has been transmitted.

For L-band scatterometers the calibration tone appears in double sideband form on the cosine channel at 1900 Hz. The polarization tone is inserted similarly at 2100 Hz. The pointers are

$$PCC = \left[1900 \left(\frac{512}{f_s}\right) = 0.5\right]_I \tag{8.39}$$

and

$$PCV = \left[2100 \left(\frac{512}{f_s}\right) + 0.5\right]_I \tag{8.40}$$

for calibration and polarization respectively. When no tone at PCV appears (compared to noise level) then horizontal polarization has been transmitted.

Noise floor pointers may be calculated for C-band or L-band respectively by

$$PNZ_C = [1800\ (\frac{512}{f_s}) + 0.5]_I \tag{8.41}$$

$$PNZ_L = [1200\ (\frac{512}{f_s}) + 0.5]_I \tag{8.42}$$

Step 7 - It is quite possible that Interrupt-1 will occur prior to completing Step 6.1 of SIGMA1. Therefore, the module should be coded to handle such an event. The flow charts in Appendix F indicate a way to process the interrupt and alert the operator to that fact. This capability is designed to be used in the program development and testing only.

SIGMA2 (Additional considerations):

Step 2 - This step should accomplish the following:

a. Evaluate power at the pointer PCH and PCV if C-band (or PCC and PCV if L-band).

b. Evaluate power at the noise floor, PNZ, Equation (8.41) or (8.42). Reset calibration alarm.

c. For C-band scatterometers:
   1) If PT_H (power at PCH pointer) >> PT_V (power at PCV pointer) set ^{C}(PZFLG) = 2 to indicate horizontal polarized transmitter.
   2) If $PT_H << PT_V$ set ^{C}(PZFLG) = 0 to indicate vertical polarized transmitter.
   3) If ^{C}(PZFLG) = 2 compare PT_H to PT_N (power at PNZ pointer) and if $PT_H >> PT_N$, go to C.(5), else set calibration alarm and go to C.(5).
   4) If ^{C}(PZFLG) = 0 compare PT_V to PT_N and if $PT_V >> PT_N$ go to C.(5), else set calibration alarm.
   5) Continue, transmitter type and quality have been found.

d. For L-band scatterometers:

1) If PT_C (power at PCC pointer)>>PT_N go to d. (2), else set calibration alarm.

2) If PT_V (power at PCV pointer)>>PT_N set ^{C}(PZFLG) = 2 to indicate horizontal polarization and go to d (3), else set ^{C}(PZFLG) = 0 to indicate vertical polarization.

3) Continue, transmitter type and quality have been found.

Step 2 - Total received power within each band associated with θ_{Di} is calculated by

$$PR_i = \sum_{k = PRL_i}^{PRL_i+N_i-1} DSL_k \qquad (8.43)$$

where DSL_k is the magnitude squared of kth spectral line coefficient of the 512 point DFT. In the CZT implementation of the DFT the Fourier components are squared and integrated over N_R sub-records in a dedicated hardware processor, so that only the summation over the

$$NF_i = PRR_i - PRL_i + 1 \qquad (8.44)$$

set of spectral lines is required to estimate the returned power.

Actual transmitted power is found by correcting for the fact that the calibration tone may not appear in the center of the filter bandwidth. The measured transmitted power is adjusted by the shape function in Equation (8.21).

When using gain and beamwidth look-up tables the values of θ'_{Li} will be used to index into the tables, rather than the "desired" viewing angles, θ_{Li}.

Step 4 - All data except flag bytes going into the output buffer lines must first be converted to binary coded decimal (BCD).
Appendix F shows the line format.

Step 5 - As each output buffer line, indexed by KO, Equation (8.34), is written to the Bi-ϕ L port it should be backfilled with BCD '9' in each byte.

8.2.6 Output to the Bi-ϕ L Interface

Output to the Bi-ϕ L interface is accomplished by transferring the contents of the block of memory, one byte at a time, to the Bi-ϕ L output port. The software used in the 13.3 GHz processor built at TAMU in 1977 may be used with appropriate modifications to account for the differences in the BCD data frame.

8.2.7 Estimated Core Memory Usage

Expected core usage may be divided into five principal blocks as illustrated in Figure 8.11. This breakdown does not imply specific address spaces for the modules involved. The size of each usage space has been conservatively estimated and is expected to be somewhat less, but of course will only be determined when actual coding is accomplished.

Block 1 is a 1K segment set aside to handle all hardware related address ports; e.g., Front Panel, Bi-ϕ L, NERDAS, and 8259 (Interrupt Processor).

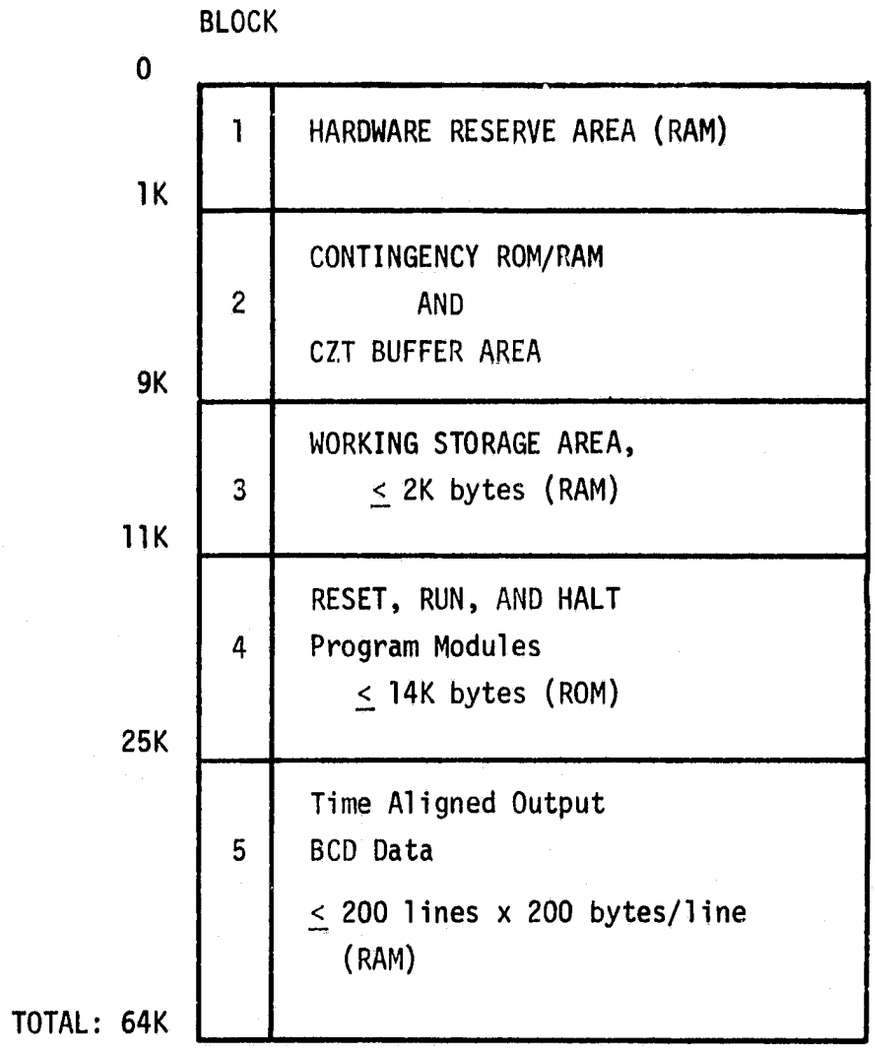


FIGURE 8.11 ESTIMATE CORE USAGE

Block 2 is contingency space containing both RAM and ROM, up to 8K bytes.

Block 3 is a set of RAM reserved for working storage; i.e., space required for data manipulation (sigma-zero calculations), NERDAS decoding, Bi-ϕ L encoding, input CZT data, etc.

Block 4 is a 14K byte set of ROM reserved for the main driver program; i.e., RESET, RUN, and HALT module.

Block 5 is a 39K byte region of RAM required for the BCD Output data set containing sigma-zero values aligned with NADIR time tags. This buffer is large enough to allow aircraft operations up to altitudes of 6500 feet above the terrain being viewed.

8.3 Operating Procedures

8.3.1 Set-Up

Consistent with the design goals of the system, minimum set-up operations will be required. Operator interactions consist of only two functions:

a. Power - Turn-On: assure that the system has proper electrical power and that In-Put/Output ports are properly connected and powered.

b. Enter any NERDAS over-ride parameters.

Over-ride parameter inputs, as discussed in Section 8.2.2, must be entered before placing the system in the "RUN" state.

8.3.2 Operator Inputs

The operator may communicate with the system through the keyboard

only. The keyboard will have three state keys (RESET, RUN, HALT), eleven digit keys (0,1,...,9'.' CHS) and three function keys (CLR, DISP, ENTER). The state keys are used in place the system in the desired mode (or state) of operation. When power turn-on is accomplished the system is automatically placed in the RESET state. While in this state the operator should enter those aircraft data values, if any, that will be used in place of NERDAS values. After the over-rides are entered, they may be verified by the display function (DISP key). If during entry of value on the keyboard a key error (0,1,...,9,'.', CHS) is made, that entry may be cleared from the board by pressing the CLR key. The appropriate sign may be given to the value by pressing the CHS key. Once the value desired is placed in the keyboard display, the operator may implement the over-ride by pressing ENTER and then the destination key; i.e., VEL, ALT, DRIFT, ROLL or PITCH. Procedural errors during over-ride entries and during display requests are signaled by flashing the front panel display. Return to the RESET mode is then made by pressing the CLR key. Table 8.2 summarizes the operator input procedures in their correct sequence.

Only the one error message (flashing display) is provided to the operator while in the RESET state. This always signifies an invalid key sequence during data entry on display functions.

System defaults are needed for those values of the aircraft data set which have not been over-ridden, but fail to pass a reasonable value test in the NERDX module. It is suggested that these default values be

TABLE 8.2 OPERATOR INPUT SEQUENCE

Step 1. Power--on and I/o connections made.

Step 2. Over-ride Values:

a. If no over-rides desired go to Step 3.

b. Key in value and sign, (floating point number)

Example: (key order is left to right)

"-0.19", displayed shows - 0 . 1 9

press "ENTER", display blinks

press "ROLL", display shows

R 0 L = - 0 . 1 9

c. Press CLR to blank front panel display.

d. Key in next value; as in 6 above, until all desired over-rides entered.

e. Sequentially display all values previously entered,

Example: press "DISP", display blinks

press "ROLL", display shows

R 0 L = - 0 . 1 9

Step 3. Change to "RUN" state by depressing RUN key.

Step 4. Change to "HALT" state by depressing HALT key.

Step 5. Change to "Reset" state by depressing RESET key.

selected based upon the most likely operating envelope for the system; e.g., ALT=1500', VEL=77.1 mps, PITCH=ROLL=DRIFT=0.0.

8.3.3 System Outputs

8.3.3.1 The Output Data Frame

Each cycle through the SIGMA1/SIGMA2 modules will cause an output buffer line, or 'frame', to be written to the Bi-ϕ L interface. Each frame will contain up to 200 bytes of BCD data and flags. As the frame is now designed, pages 47-50 of Appendix F, it will consist of 128 (80H) bytes. All data will be coded into BCD except the flag words 78H through 7FH.

8.3.3.2 Data Frame Annotation

Output flags are place in the frame to provide the user with sufficient data to analyze sigma-zero results. These are information bits which indicate specifically the conditions and over-rides, if any, used in calculating the sigma-zero estimates.

Each output frame will contain the following trailer bytes ('>' indicates binary notation).

Polarization: >11 = HH,>10 = HV,>00 = VV,>01 = VH
H/V Alarm: >00 = off,>01 = on
Time Alarm: >00 = off,>01 = on
Calib. Time Alarm: >00 = off,>01 = on

Over-ride Flag: 00H = No over-rides used
Bit 6 on = alt. over-ride used
Bit 5 on = vel. over-ride used
Bit 4 on = pitch over-ride used
Bit 3 on = roll over-ride used
Bit 2 on = drift over-ride used

Std-value Flag: (same format as over-ride)

9.0 CONCLUSIONS AND RECOMMENDATIONS

9.1 Executive View

The CZT approach represents a novel and powerful technique for filtering the fan beam scatterometer return into many parallel Doppler channels. When the radar return is regarded as a complex valued signal, the CZT, when appropriately modified, is capable of filtering the fore and aft spectra simultaneously. When amalgamated with a dedicated digital post processor, the PSD of the radar return can be estimated with precision over a large dynamic range as required by the character of the radar return. Since many parallel spectral estimates are available simultaneously the filtered data can be selected adaptively with changes in aircraft velocity to track the desired viewing angles.

Since the selection of spectral data is under software control, the mode of operation can also be altered to satify the needs of specific experiments. As a consequence, the spectral data could be selected to provide constant precision estimates, constant ground resolution estimates, constant angular resolution estimates or combinations of these modes. The operating system developed within this effort is based on a constant angular resolution mode. An angular resolution of three degrees is provided for the C band system and six degrees for the L band system. Higher or lower angular resolutions are obtainable for both scatterometers.

The charge coupled device and associated circuitry were demonstrated as a good approach to filtering the radar data. A dynamic range of

more than 50 dB was observed in the evaluation tests. Excellent linearity was demonstrated over the 50 dB of dynamic range. The quadrature processing technique yielded a fore to aft signal rejection ratio of more than 30 dB.

A comprehensive software design was developed to control the processing system and to convert the spectral estimates into calibrated estimates of σ°. The software was designed into flexible modular units having cohesive objectives. Flags were included in the design to assist in testing the program once the coding is developed. An identical software architecture was utilized for the C-band and L-band processors. The same architecture is extendable to other NASA fan beam scatterometers.

Various interface units were also designed to provide communication in and out of the system. Among these, an operator console and display unit was designed to permit operator control of the system. The console features a keyboard entry and an alpha-numeric display. The display incorporates many intelligent functions such as clear display, shift left or right, blink, read/write, etc. It consequently represents a powerful technique to monitor the operation of the processing system when used appropriately. For example, it will be helpful in debugging the software during the future development stages.

An aircraft interface unit was also designed to transform NERDAS data to a binary form for use by the scatterometer processing system. A similar interface unit was actually fabricated and proven under another NASA contract [1] and features a level control to condition the Bi-ϕ L modulation for decoding and a special digital decoder

to properly identify the binary transitions.

The afore described processing system consequently meets or exceeds the requirements for a flexible operator and experimentor oriented system. The system architecture is sufficiently universal to handle the data characteristics unique to each of the NASA scatterometers. It is conceivable that a single processor could adapt to each of the scatterometers, if desired. However, it is recommended that a processor be developed for each of the scatterometers to provide multi-sensor data simultaneously. In the latter case, since a common design approach applies to all the systems, the development and fabrication costs will be considerably cheaper.

It is important to point out that the efforts reported here represent a comprehensive paper design with certain elements of the design partially proven through breadboard testing. This work is consequently a sizable first step in a design and development program. However, a major effort is yet required to fully develop the design since a design and development program is an iterative process including design, fabrication and evaluation.

The results of the current effort have demonstrated that it is necessary to develop and evaluate an engineering model of the processor system before fabricating the target system (see Section 6.0). An evaluation of the engineering model is required to assess the actual speed of the processor and the interaction among all elements of the system. In approaching the development of the engineering model, it is recommended that certain key problem areas, identified within this

effort, be investigated before implementing the developmental system.

First, it is necessary to perform a comprehensive study to verify the current σ° inversion technique or to replace it with a better one. The current inversion technique is based on a intuitive interpretation of the scatterometer equation and has not been verified through simulations. It is essential that this be performed to assure accurate σ° estimates accounting for the effects of the antenna pattern, Doppler filter and aircraft attitude. Secondly, the software design should be encoded to verify timing, optimize interrupts and develop certain interactive coding for debugging purposes and system check-out. Lastly the breadboarded PSD estimation subsystem should be employed as a "test bed" to aid in the development if the engineering model. In this regard the source of the noise floor problem described in Section 6.5.3 should be identified and remedied. Then the software and hardware should be united to demonstrate system capability.

9.2 Engineering View

Experience in debugging and evaluating the PSD estimation breadboards has led to a number of insights which could lead to improvements or simplifications in the system. They are simply listed below:

1) Consider digitization to 8 bits rather than 12. The additional resolution may be available in averaging records and spectral lines. The accumulator word size could conceivably be reduced to 24 bits.

2) Consider reducing memory size to store only aft data.

3) Use doubly clad PC board to improve noise immunity of analog and A/D electronics.

4) Consider methods of shielding the filtering unit from the digital and controller boards.

5) Replace discrete sample and hold circuits with integrated circuits.

6) Establish the methodology to balance the common mode rejection in the differential current integrators.

7) Evaluate the effect of op amp offset voltages on pre-chirp multiplication accuracy.

8) Reduce the bandwidth of the input amplifiers by a factor of two and compensate for the loss of calibration tow amplitude in the software.

9) Optimize CZT gating signals to reduce probable noise sources.

REFERENCES

[1] "Definition and Fabrication of an Air borne Scatterometer Radar Signal Processor," Technical Report RSC3182-2, Remote Sensing Center, Texas A&M University, NASA Contract NAS9-14493, December 1976.

[2] Reeves, R. G., et al, Manual of Remote Sensing, American Society of Photogrammetry, 1975.

[3] Claassen, J. P., "Accuracy Criterion for Estimating the Mean Squared Signal," RSL-TM 186-1, Remote Sensing Laboratory, University of Kansas, 1970.

[4] Oppenheim, A. V. and R. W. Schafer, Digital Signal Processing, Prentice - Hall, Inc., 1975.

[5] Claassen, J. P., "The Design of the RADSCAT Experiments," RSL-TR-186-2, Remote Sensing Laboratory, University of Kansas, 1971.

[6] Wait, J. V., L. P. Huelsman and G. A. Korn, Introduction to Operational Amplifier Theory and Application, McGraw Hill Book Company, 1975.

[7] Brodersen, R. W., C. R. Hewes, D. D. Buss, "A 500 Stage CCD Transversal Filter for Spectral Analysis," IEEE Transactions on Solid State Circuits, Vol. SC-11, No. 1, pp. 75-84, February 1976.

[8] Hill, F. J. and G. R. Peterson, Digital Systems: Hardware Organization and Design, New York: John Wiley and Sons, Inc., 1973, p. 85.

[9] Rolander, T., "Intel Multibus Interfacing," Santa Clara: Intel Corporation, 1977.

[10] "Feasibility Study of Microprocessor Systems Suitable for Use in Developing a Real-Time Processing For the 4.75 GHz Scatterometer," Final Report RSC-3556-1, Remote Sensing Center, Texas A&M University, August, 1977.

[11] "AMC 95/4000 Monoboard Computer," Santa Clara: Advanced Micro Computer, 1978, p. 2.

[12] "MTX-A1," Montreal: Matrox Electronic Systems, 1977.

APPENDIX A

APPENDIX A

OPTIMIZING THE PRECISION FOR A GIVEN COVERAGE

Theorem:

It is shown here that for a given coverage interval

$$L_c = \rho_B + vT$$

that the precision factor BT can be maximized if the scan time is chosen so that

$$\rho_B = vT$$

ie., the beam resolution is equal to the scan length. As a consequence, BT can be maximized if the beam resolution occupies half the coverage length during a single integration period. The reader will note that the result is independent of view angle.

Proof:

The Doppler frequency on the ground track is given by

$$f = \frac{2v \sin\theta}{\lambda} .$$

When the x coordinate is chosen in the up-track direction as illustrated in Figure A-1, the Doppler gradient along the x coordinate is given by

$$\frac{df}{df} = \frac{2v}{\lambda} \quad \frac{1}{n \sec^3\theta}$$

where $x = h \tan\theta$ has been used. For a fixed bandwidth the precision factor may be related to the gradient in the following way

$$BT = \frac{df}{dx} \rho_B \frac{(L_c - \rho_B)}{v}$$

The factor $(L_c - \rho_B)/v$ is clearly the integration time. Now the precision can be maximized with respect to beam resolution by determining when

$$\frac{dBT}{d\rho_B} = 0$$

This yields the result

$$\rho_B = \frac{L_c}{2}$$

which is equivalent to

$$\rho_B = vT$$

It is easily shown that this criterion results in a maximum BT rather than a minimum.

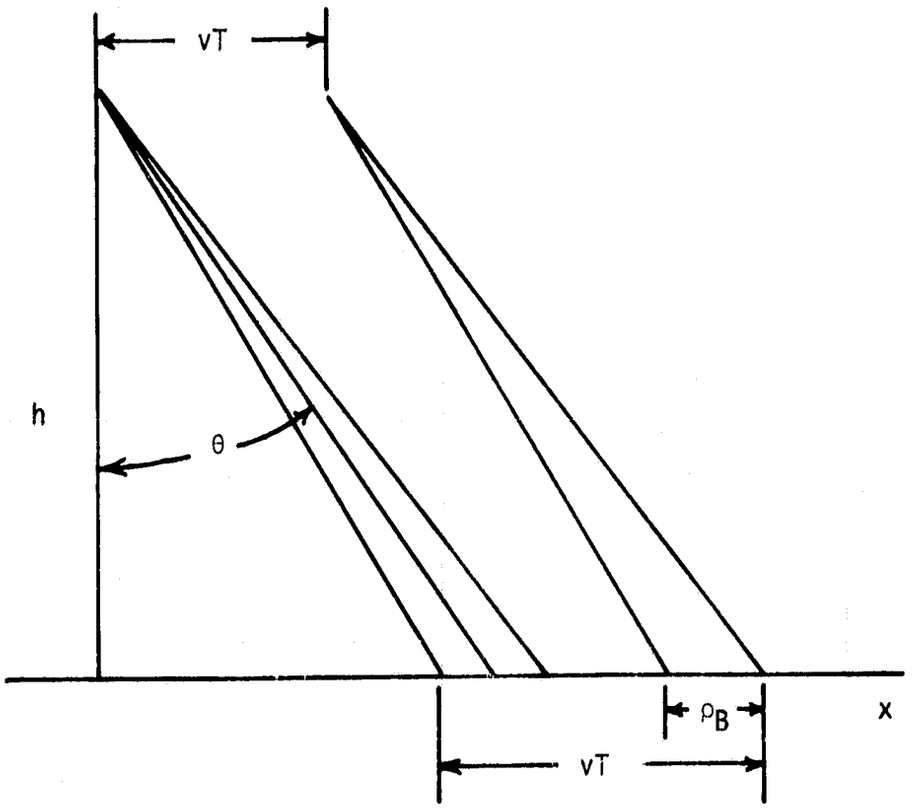


FIGURE A-1 PARAMETERS REVELANT TO THE PROOF

APPENDIX B

APPENDIX B

THE THEORY AND APPLICATION OF THE ALIGNMENT GENERATOR

The alignment generator produces quadrature periodic square wave signals given by

$$V_I = \begin{cases} -1 & -T/2 < t < -T/4 \\ +1 & -T/4 < t < T/4 \\ -1 & T/4 < t < T/2 \end{cases}$$

and by

$$V_Q = \begin{cases} 1 & 0 < t < T/2 \\ -1 & T/2 < t < T \end{cases}$$

over a single period. As can be easily shown, the spectra associated with these periodic signals are given by

$$S_I = \sum_{\substack{n=-\infty \\ n \text{ odd}}}^{\infty} \pm \frac{1}{n} \delta\left(\omega - \frac{2\pi}{T} n\right)$$

and

$$S_Q = \sum_{\substack{n=-\infty \\ n \text{ odd}}}^{\infty} \pm \frac{1}{n} \delta e^{j\frac{\pi n}{2}} \delta\left(\omega - \frac{2\pi}{T} n\right)$$

C-3

respectively, where 1/n is negative on the set $n\epsilon$ {..., -7, -3, +3, +7,...} and positive on $n\epsilon$ {..., -5, -1, +1, +5, ...}. S_I and S_Q are illustrated in Figure B-1. If V_I is inserted into the I channel and V_Q is inserted in the Q channel , then the CZT processor will yield a spectrum given by

$$S_o = S_I + j\, S_Q$$

The spectrum of S_o is shown in Figure B-2. It is noted there that the spectrum is not symmetric and that alternate lines have disappeared. The complementary spectrum can be formed by interchanging the inputs to the channels. Then the complex valued linear combination of spectra becomes

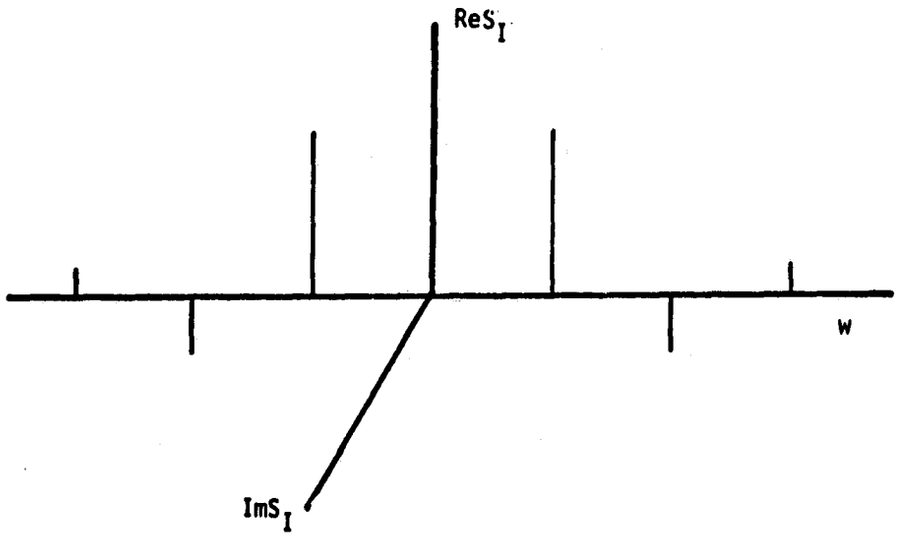
$$S_o' = S_Q + j\, S_I$$

which is illustrated in Figure B-3. It is noted that

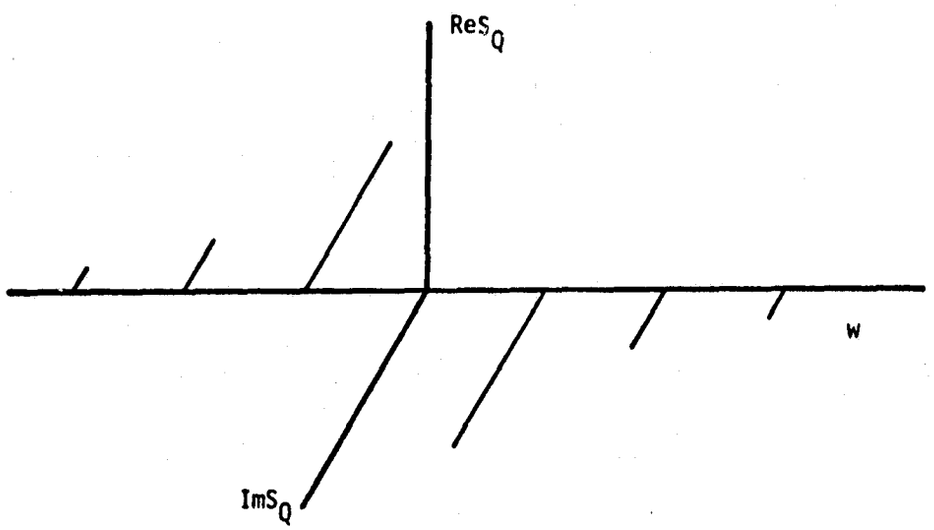
$$S_o'(\omega) = j\, S_o(-\omega)$$

i.e., $S_o'(\omega)$ is the convolved and phase rotated spectrum of $S_o(\omega)$.

The spectra of S_I, S_Q, S_o, and S_o' are helpful aligning the CZT filtering unit. The gains through the four chirping channels must be matched to assure operation of the CZT transversal filter consistent with the theory cited in Section 3.4. Before using the alignment generator, the dc offsets in the channels prior to the chirping multipliers must be removed. An offset adjustment is provided in each of the four channels.



a) Spectrum of S_I



b) Spectrum of S_Q

FIGURE B-1 COMPARISON OF SPECTRA OF V_I AND V_Q

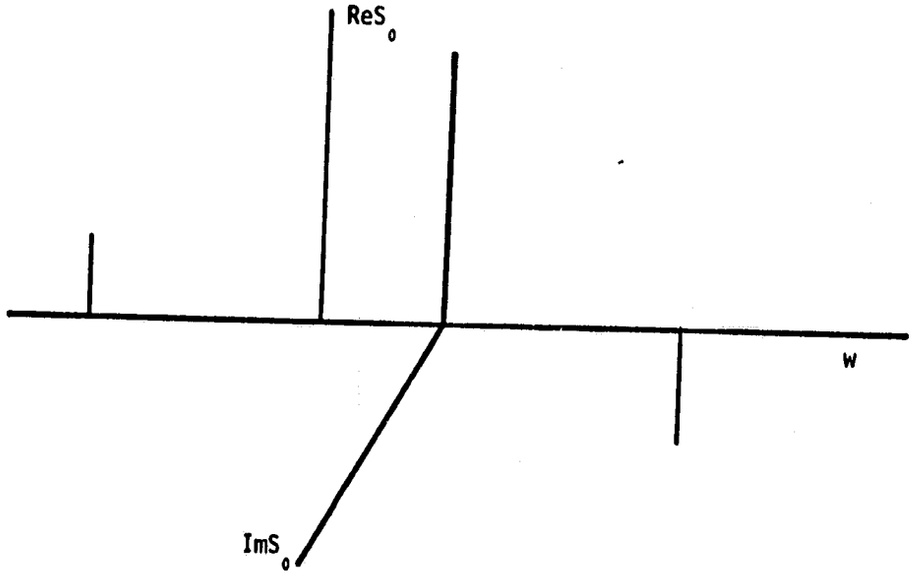


FIGURE B-2 THE SPECTRUM OF $V_I + jV_Q$

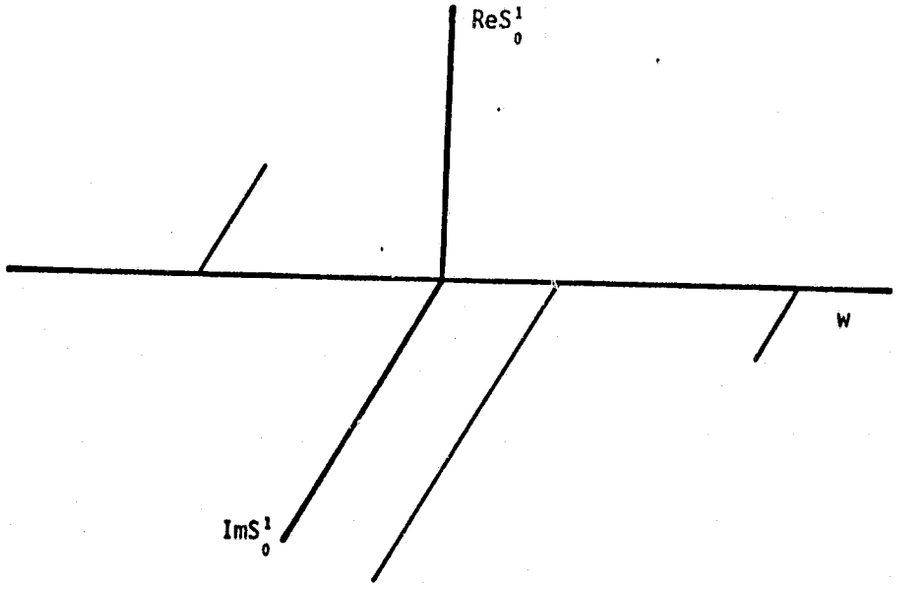


FIGURE B-3 THE SPECTRUM OF $V_Q + j\ V_I$

Gain Adjustments

Inject V_I into the I channel (x) with no input on the Q channel (y). Adjust x_1 and x_2 channel gains so that Re and Im inputs to the transversal filter have identical peak to peak amplitudes. An oscilloscope may be employed to observe the Re and Im inputs to the transversal filter. Also adjust gains in the y_1 and y_2 channels in the same way by injecting V_I into the Q channel.

After balancing the gains it is advisable to fine adjust the dc offsets in all four channels.

DC Pedestal Adjust

The chirped signals must ride on a positive bias when entering the CCD transversal filter. A bias somewhere between 4 and 9 volts is required to achieve the full dynamic range inherent to the transversal filter.

To adjust the dc pedestal inject V_I into the I or Q channel and monitor the delayed output VOR with an oscilloscope. Adjust the dc pedestal on the Im channel so that the delayed output is neither compressed or clipped at the top or bottom of the chirped waveform. For maximum undistorted dynamic range, the input to the transversal filter should be near 1.5 volts peak to peak.

Similarly, adjust the dc pedestal on the Re input by monitoring the delayed output VOI.

The envelopes of VOR and VOI will be amplitude modulated over the CZT cycle. This is normal for the CCD device and is induced by a high

frequency roll-off inherent to the filter. The filter taps have been compensated for this characteristic.

Alignment Verification

Before verifying the alignment of the CZT filtering unit, the dc offsets in the signal extraction channels should be removed. This may be accomplished by adjusting VO1 and VO3 to zero volts dc when no inputs to the I and Q channels are applied.

With the dc offsets removed, inject V_I and V_Q into the I and Q channels, respectively. To stabilize the spectral lines appearing at VO1 and VO3 it is advisable to synchronize V_I and V_Q with the CZT clock. This may be accomplished by driving the alignment generator at a sub-harmonic of the sampling clock. The outputs VO1 or VO3 or both should contain a spectrum identical to that of Figure B-2 when monitored with an oscilloscope. The oscilloscope gain may be increased to examine the intensities of the suppressed lines. If they are present, then the CZT filtering unit is not properly aligned. The location of the suppressed lines may be ascertained by removing the Q or I input. If the output spectrum matches that of Figure B-3, then the input signals have been interchanged.

APPENDIX C

APPENDIX C

THE DESIGN AND OPERATION OF THE ALIGNMENT GENERATOR

In order to emmulate quadrature radar return signals, a signal alignment generator was designed and constructed to produce quadrature periodic square wave signals. The circuit in Figure C-1 was utilized for that purpose.

The incoming square wave is sent directly to the clock input of one D type flip flop (FF) and, at the same time, is complimented by use of a NAND gate and sent to the clock input of the other D FF. Due to the fairly low frequency of the system input, the delay associated with the NAND gate can be neglected. The clear and preset inputs to both FF's are disabled by typing them high. Since a D FF is triggered by a rising edge on the clock cycle, the effective output frequency is one half of the input frequency. The clocking signals to the D FF's are 180 degrees out of phase, which yields signals out of the divide-by-two circuitry that are 90 degrees out of phase. This is demonstrated in the timing diagrams in Figure C-2. The top signal is the input square wave (CL1) from the controller board. The second waveform is the complementary clock signal (CL2) from the output of the NAND gate. The third and fourth waveforms are the outputs (Q_i and Q_q) of the in-phase and quadrature FF's respectively. Note that the outputs only change on the rising edges of the clock inputs.

By connecting the complementary output ($\overline{Q}_i$) to the input of the in-phase FF, the in-phase FF will operate in a toggle mode. The complementary output of the in-phase signal ($\overline{Q}_i$) and the quadrature signal (Q_q) are

NANDed together and the resulting output is used as the input to the quadrature FF. The output of the NAND gate only goes low when both inputs are high. For that case to be true, the quadrature signal (Q_q) must be high and the in-phase signal (Q_i) must be low (state A). With the output of the NAND gate in the low state, the next CL2 rising edge will cause the quadrature FF to toggle into the low state (state B). This change in the quadrature output to the low state will cause the output of the NAND gate to go high. One half cycle late, CL1 will produce a rising edge which toggles the in-phase outputs (state C). The change will not affect the output of the feedback NAND gate therefore the input to the quadrature FF will remain in the high input (state D) and the resulting output will cause the output of the NAND gate to go low again (state E) after the next CL1 rising edge and the cycle begins again. The input to the quadrature FF can be seen in the fifth waveform displayed in Figure C-2. This arrangement will assure that the quadrature (SIN) channel reference signal will always lag the in-phase (COS) channel by a quarter period.

The output signals from the FF's are buffered by the diode, transistor, and resistor circuits before being converted to analog signals by the LM301 operational amplifiers (op amps). By careful selection of the resistors used in the op amp circuits, the output analog signals can be balanced.

In order to provide stable spectral lines at the output of the CZT filter, the alignment generator should be synchronous with a sub-harmonic of the CZT clock signal. This signal is obtained by tapping outputs from the CZT clock signal counter (ZG1) on the controller board. For ease in

presentation on the oscilloscope, the fifth sub-harmonic (1/16 of the CZT clock rate) was utilized (pin 7).

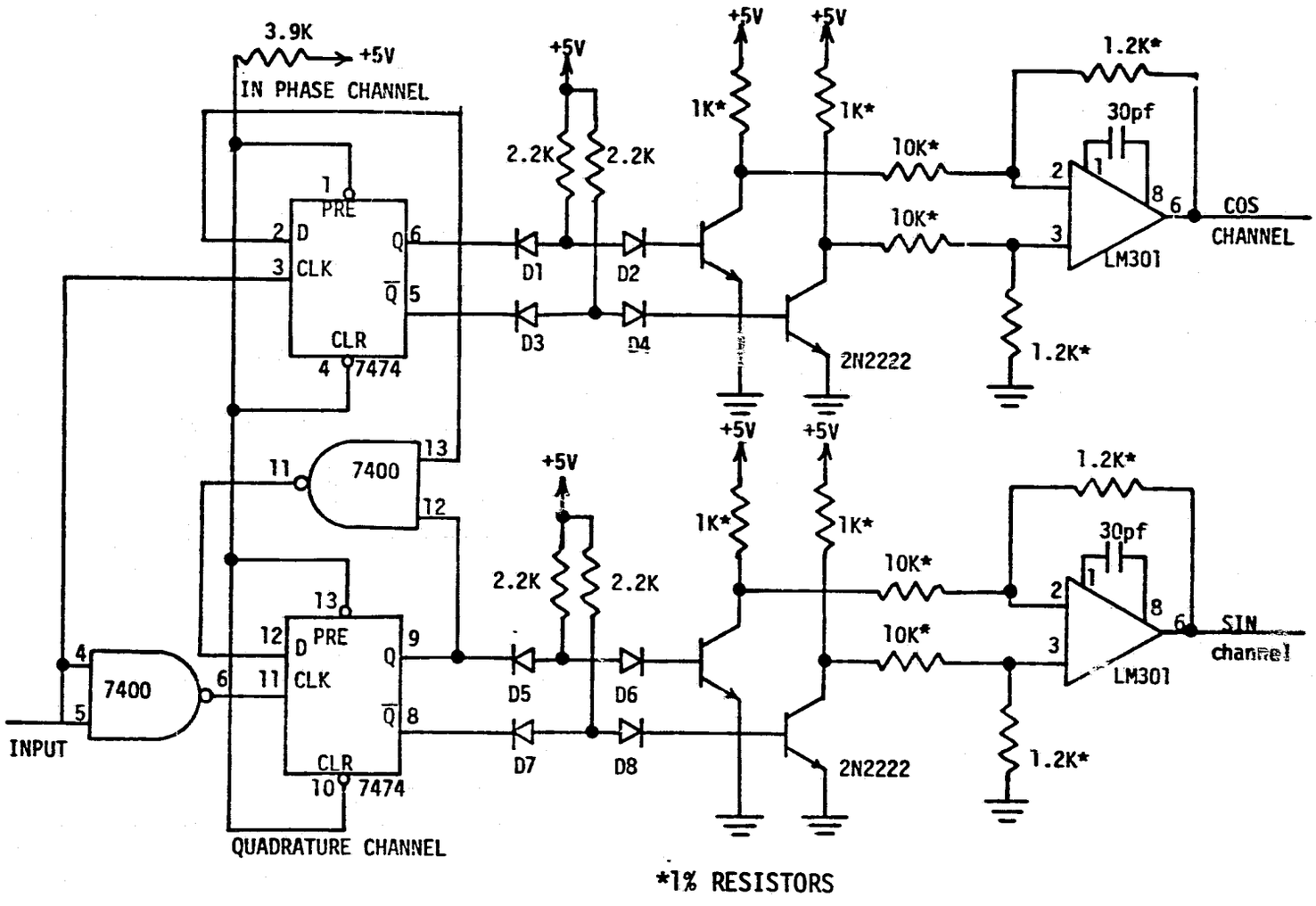


FIGURE C-1 SCHEMATIC OF ALIGNMENT GENERATOR

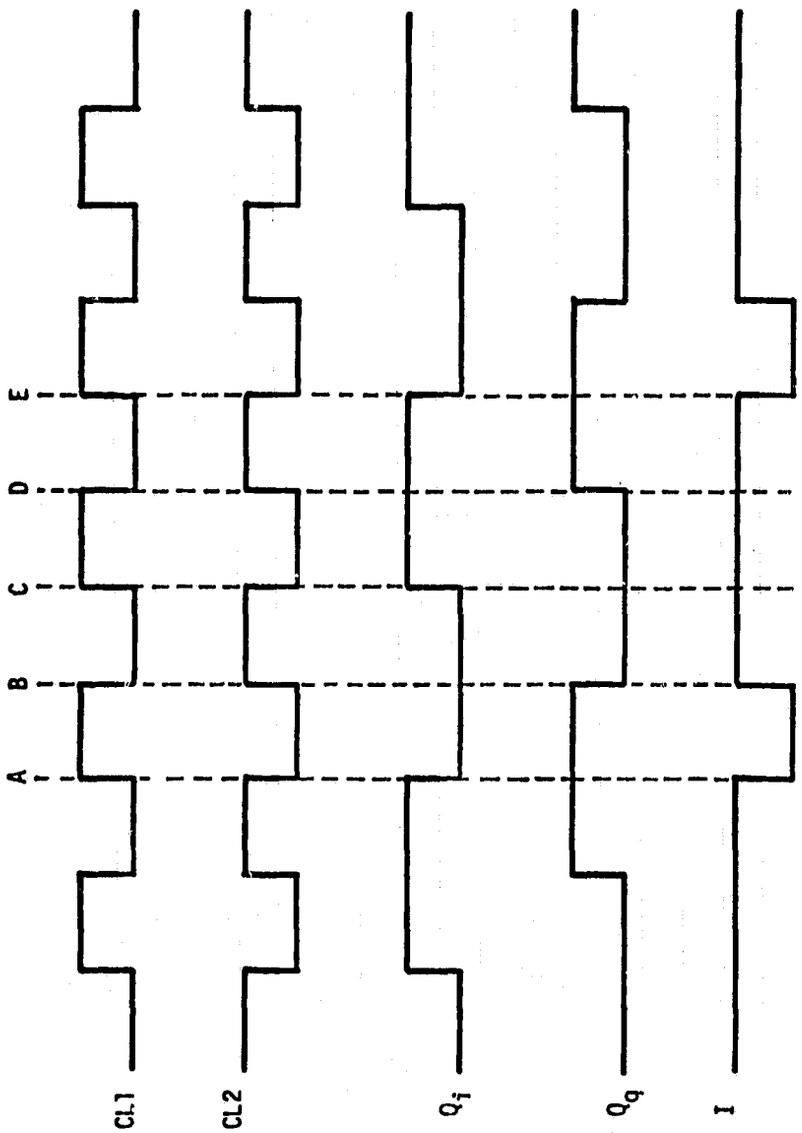


FIGURE C-2 TIMING DIAGRAMS

APPENDIX D

APPENDIX D

PARTS LIST FOR SCATTEROMETER SIGNAL PROCESSOR

I. Controller Board

Number	Device
8	SN7474N Dual D flipflop, TI
7	SN7408N Quad 2-input NAND Gate, TI
15	SN74193N Binary up-down counter, TI
2	316A102 Allen Bradley resistor package
2	C2708 EPROM INTEL
3	SN74174N 4 bit bidirectional shift register, TI
2	SN7417N Hex Buffer/Driver, TI
3	SN7432N Quad 2-input OR gate, TI
7	SN7404N Hex Inverter, TI
1	SN7430N 8-input NAND gate, TI
1	P8205 1 of 8 binary decoder, INTEL
2	SN74273N Octal D flipflop w/clear, TI
3	P8212 8 bit input/output port, INTEL
1	SN7411N 3 input positive AND gate, TI
1	SN74LS112N Dual J-K flipflop (neg. edge), TI

II. Digital Board

Number	Device
16	P2111 RAM, INTEL
5	SN7474N Dual D flipflop, TI
8	P8216 4 bit bidirectional bus transceivers, INTEL
8	SN74273N Octal D flipflop w/clear, TI
8	SN7408N Quad 2-input AND gate, TI
4	SN74LS352N Dual 4-line to 1-line data selectors, TI
14	SN74283N 4-bit binary full address, TI
5	SN7404N Hex inverter, TI
2	ADC-HS12BGC Datel A/D Convertor
2	MPY-12AD TRW 12 x 12 Multiplier
4	64x503 Spectrol Trim Pot 20 turn 50K
2	18M Resistor
2	1M Resistor
3	1K Resistor

* Unless otherwise specified:
All resistors ¼ watt 5%
All Capacitors CTC mono cap 10%

III. Analog Board

Number	Capacitors
27	0.1 mf
18	33 pf
17	0.01 mf
12	0.001 mf
5	5 pf
4	560 pf
8	470 pf
4	470 pf silver mica 50V 5%
4	420 pf silver mica 50V 5%
4	39 pf
2	1.0 pf electrolytic hermatically sealed
2	330 pf
2	1500 pf
2	10 pf
4	82 pf

Number	Resistors
4	56K
4	39K
6	21K 1%
4	12.4K 1%
4	18K
8	2.7K
6	14.7K 1%
4	8.66K 1%
4	39K
3	200K
2	1500Ω 1%
15	1K 1%
1	680Ω
4	5.1K
2	10M
15	10K 1%
4	6.8K
4	1780Ω 1%
4	820Ω
2	1.2K
1	470Ω
2	150Ω
1	100Ω
1	47Ω
1	200Ω 1%
2	5K
2	1M
2	2.32K 1%

III. Analog Board (continued)

Number	Resistors
2	7.5K 1%
2	8.2K
4	100K
2	4.7K
1	536Ω
10	.51Ω

Number	Trim Pots
6	64x502 Spectrol 20 turn 5K
4	64x103 Spectrol 20 turn 10K
2	64x102 Spectrol 20 turn 1K

Number	Integrated Circuits
18	LM 308A Operational Amplifier, TI
15	LF 356P Low Offset Monolithic JFET Input, TI Operation Amplifier
2	LM 301A Operational Amplifier, TI
3	P2708 or P2704 EPROM INTEL
4	DAC-HA12BC Datel D/A Convertor
4	DS0026CN 5MHz 2-Phase MOS clock driver, NATIONAL
1	R5601-1 Reticon Quad Chirp Transversal filter

Number	Transistors and Diodes
8	SD210, SIGNETICS
5	IN4448, TI
2	IN703A, TI
1	IN753A, TI
1	IN756A, TI

Display Interface

Number	Device
1	SN7404N Hex Inverter, TI
1	SN7432N Quad 2-input OR Gate, TI
1	SN7408N Quad 2-input AND Gate, TI
2	SN7417N Hex Buffer/Driver, TI
1	SN7474N Dual D Flip Flop, TI
20	AB316A102 Allen Bradley Resistor Package
20	SN74LS273N Octual D Flip Flop w/clear, TI
20	TIL305 Alphanumeric Display, TI
2	P8205 3-to-8 Decoder, INTEL
2	MTXA1 Matrox Alphanumeric Display/Keyboard Controller
1	MC14490 Hex Contact Decoder, MOTOROLLA
1	Hex Keyboard CHERRY B65 Series
4	SN74366AN Tri-State Hex Buffer, TI

APPENDIX E

APPENDIX E

DRAWING LIST FOR SCATTEROMETER SIGNAL PROCESSOR

Identification	Description
3556.01.01	Display, Section 1
.02	Display, Section 2
.03	Display, Section 3
.04	Display Driver, Right Half and Keyboard Controller
.05	Display Driver Left Side
.06	Keyboard
.07	Display Data Buffer
3556.05.011	Signal Conditioner for the I Channel (L-Band)
.012	Signal Conditioner for the I Channel (C-Band)
.021	Signal Conditioner for the Q Channel (L-Band)
.022	Signal Conditioner for the Q Channel (C-Band)
.03	S/H and Signal Splitters
.041	Sine and Cosine ROMS
.042	Cosine Chirp Premultiplier
.043	Sine Chirp Premultiplier
.044	Summing Circuitry
.05	CZT Filter
.06	$\emptyset_1$ Signal Extraction Circuity
.07	$\emptyset_2$ Signal Extraction Circuitry
3556.06.01	A/D Converters and Squares
3556.07.01	I and Q Summer

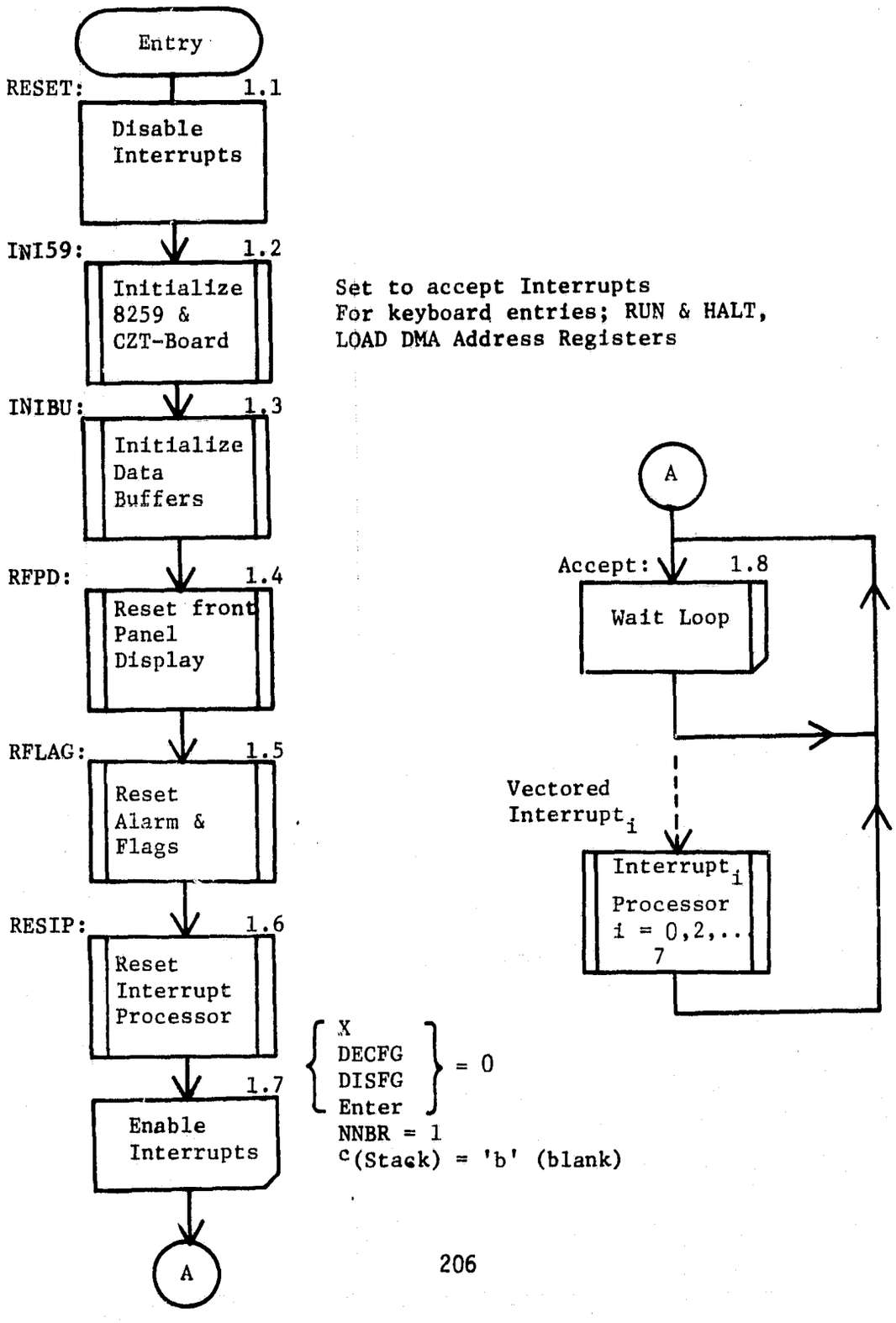
Identification	Description
3556.08.	Accumulator and Memory
.01	Preprocessor MDR and Byte Selector
.02	Preprocessor Memory Data Bus Multiplexer
.03	Preprocessor Memory Address Register
.041	Memory
.042	Memory
.043	Memory
.044	Memory
.051	Multibus Interface Logic and Buffers
.052	DMA Address Registors
.053	OD Gate, R/W Gate and E0-E7 Buffer
.054	CZT Synchronization and Gating
.06	Accumulator Adder
.07	Reset Circuitry
3556.09.01	DMA Transfer Controller
.02	DMA State Controller
.053	Clock Generator and Clock EPROM Address Register
3556.10.01	NERDAS Interface
3556.10.02	Bi-ϕ L Interface

APPENDIX F

FLOW CHARTS FOR THE PROCESSOR SOFTWARE

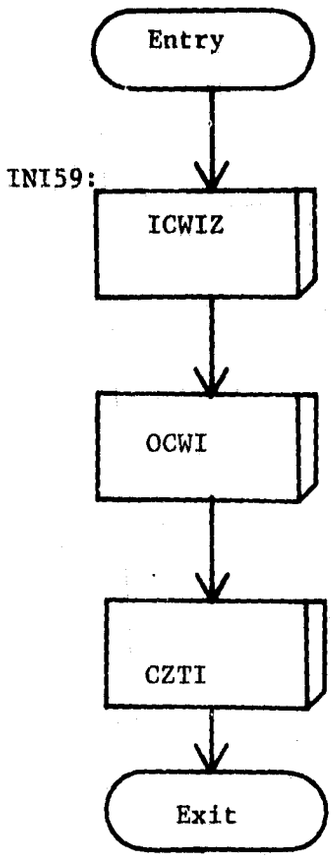
RESET MODULE

RESET MODULE, C- and L-Band Scatterometer Data Processor



RESET Module:

INI59, Initialize 8259 Interrupt Controller, set Addr. Registers



1.2.1

Issue ICWI and ICWZ to DAH & DBH,

ICW1 = A7A6A5 10110
ICW2 = A15A14...A8 = 1st VECTOR ADDRESS

Issue OCW1 to Loc. DBH,

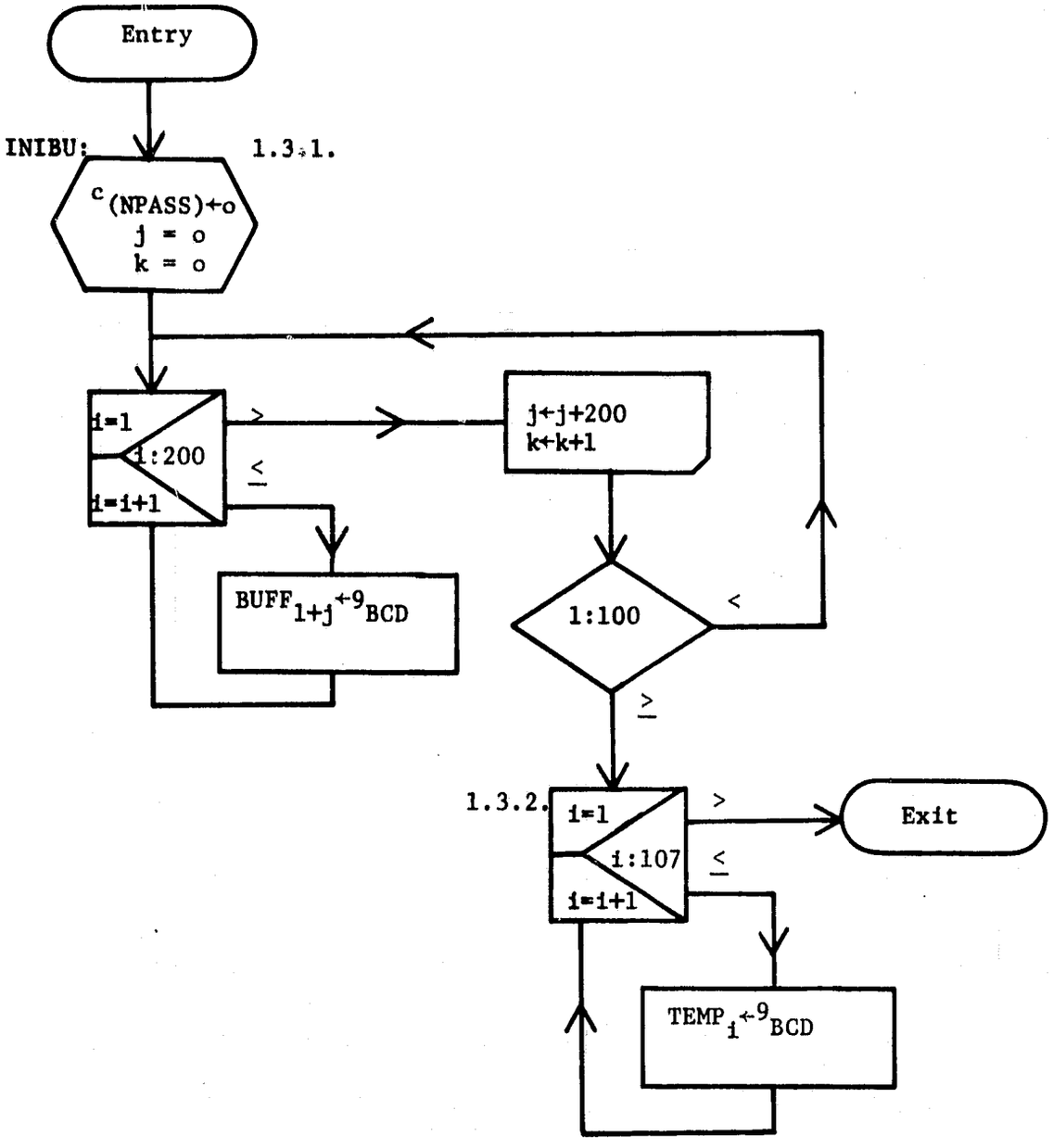
OCW1 = 02H, Masking the End-of-Convert (EOC) or Integration Timer interrupt

Initialize CZT Address Registers

Send least significant part of 'SPWR' ADDRESS to port OAH, and most significant part of ADDRESS to port OEH

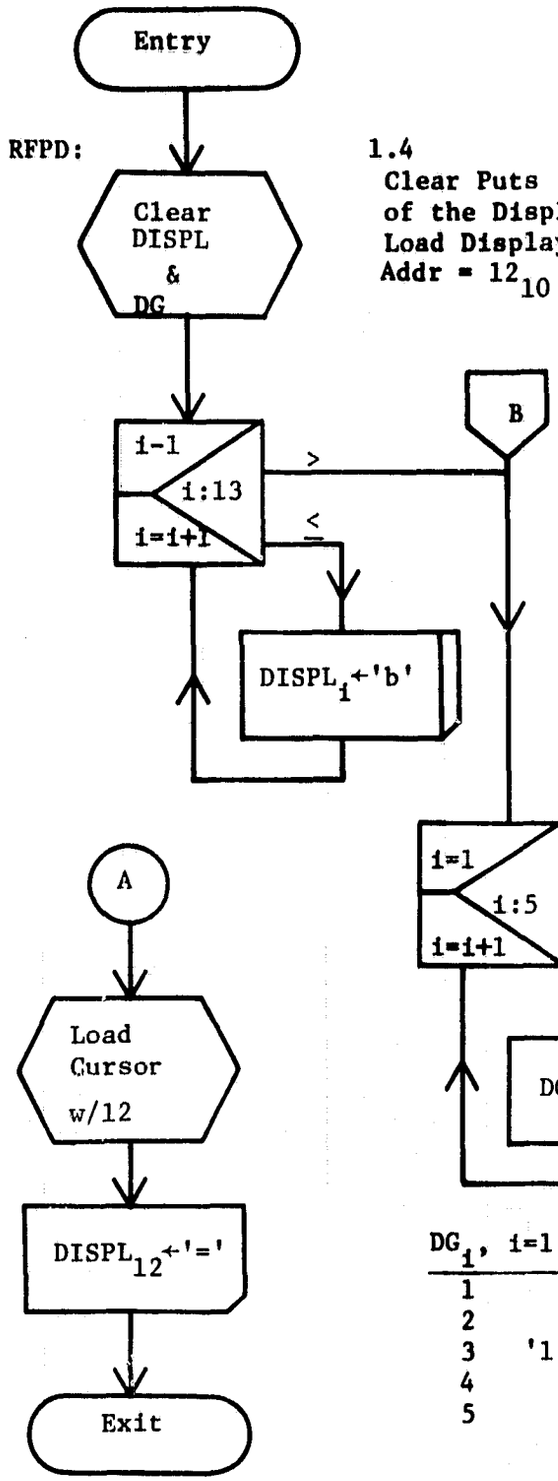
RESET Module:

INIBU, Initialize Buffers for Data



RESET Module:

RFPD, Reset Front Panel Display



1.4
Clear Puts '0' in each led
of the Display
Load Display Length Registers
Addr = 12_{10}

STATUS DISPL$_i$, i=1, 13

1	= NERDAS Status
2	= Buffer Length
3	= Time Cycle
4	= Altitude Over-ride
5	= Velocity Over-ride
6	= Pitch Over-ride
7	= Roll Over-ride
8	= Drift Over-ride
9	
10	= 3 character
11	ID Field
12	= '=' character
13	= 'b' or '-'

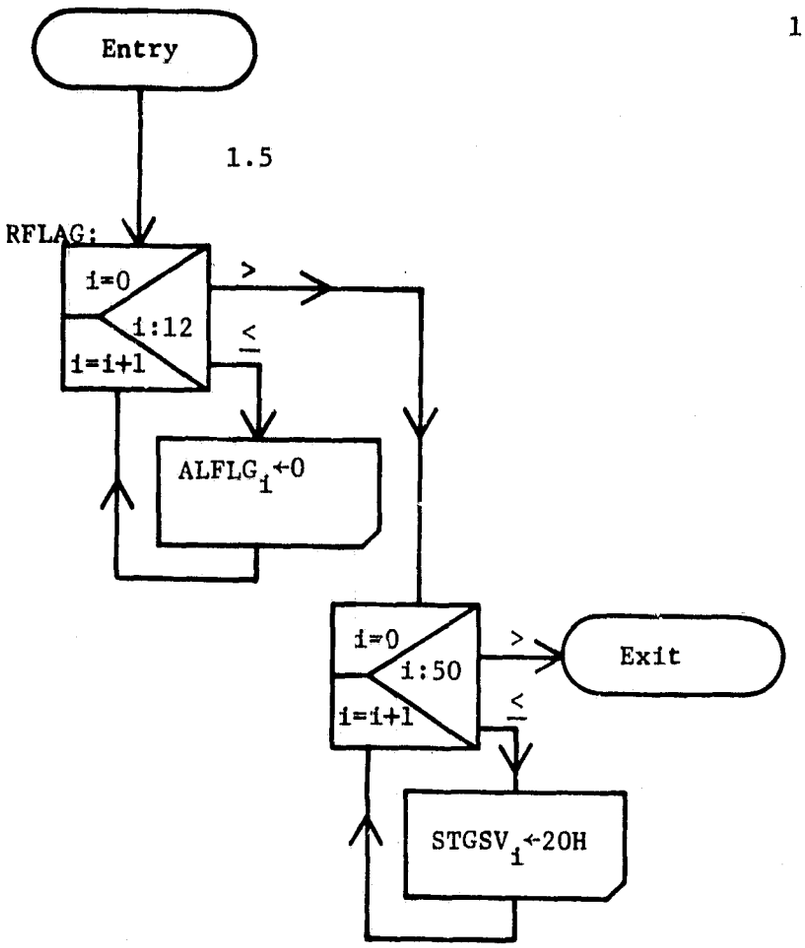
'b' (blank)
Digit Display
Addr = 14_{10}

DG$_i$, i=1, 5(Digit Displays)

1	
2	
3	'1', '2',..., '9'
4	or '.'
5	

RESET Module:

RFLAG, Reset Alarms, Flags, and Over-ride Values



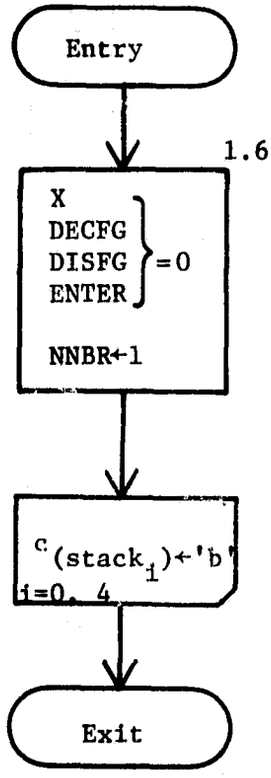
1 ← Storage Arrangement

for Alarms & Flags

```
ALFLG:   0; NERDAS
         0; Buff Length
         0; Time Cycle
         0; Alt Over-ride
         0; VEL Over-ride
         0; Pitch Over-ride
         0; Roll Over-ride
         0; Drift Over-ride
         0; Calib Err
         0; Passl Flag
 WAIT:   0; Wait Flag
 EOC:    0; Convert Flag
 LNCT:   0; Line CNTR
STGSV:20H; Character
      20H; Save Area
      20H;
        :
        :
      20H; ≤ 50 BYTES
```

RESET Module:

RESIP, Reset Interrupt Processor

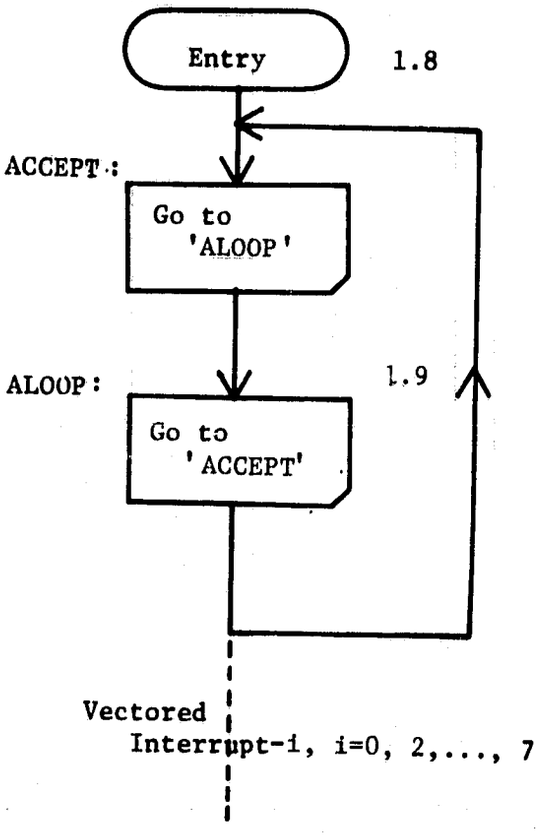


Reset Digit Counter, Decimal, Display, Enter, and Negative Flags

Fill ASCII Character Save Area with Blanks

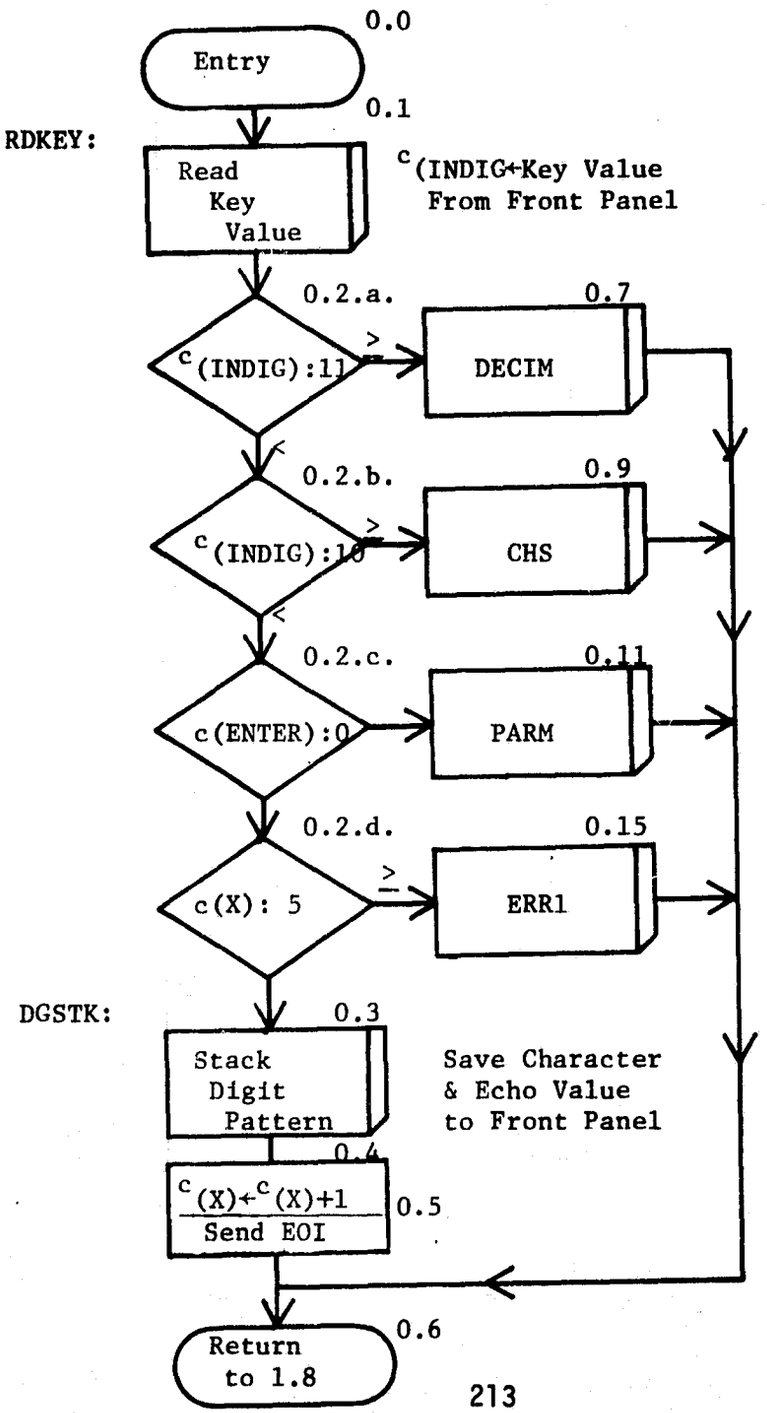
RESET Module:

ACCEPT, Wait Loop for Keyboard Entry Interrupts



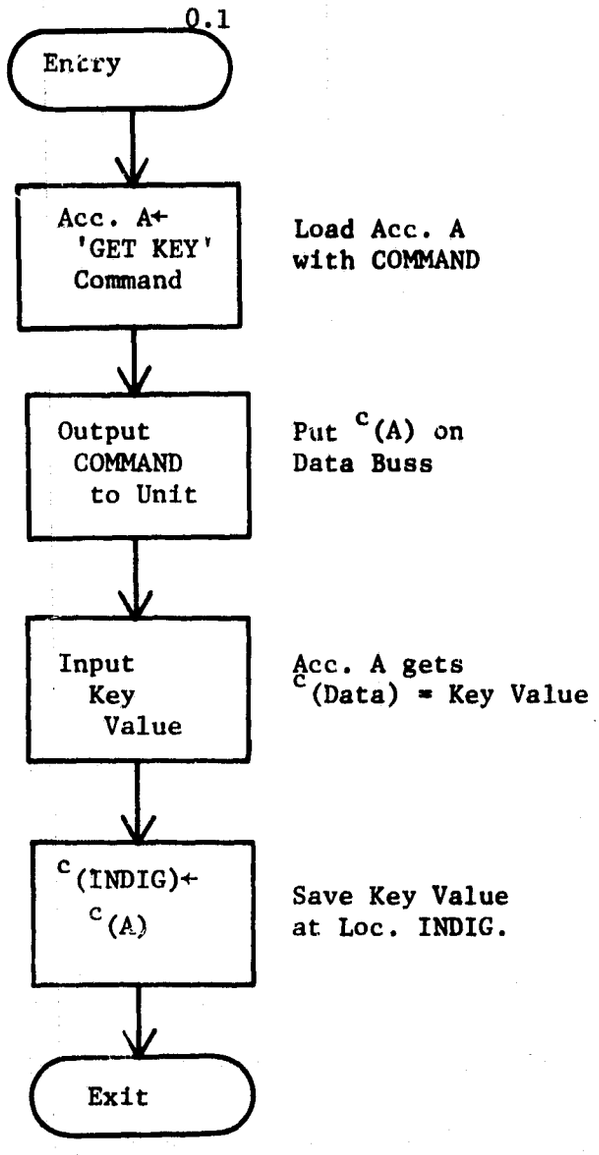
INTERRUPT PROCESSOR

Interrupt-0, Keys labelled 0, 1,..., 9, CHS, and '.'



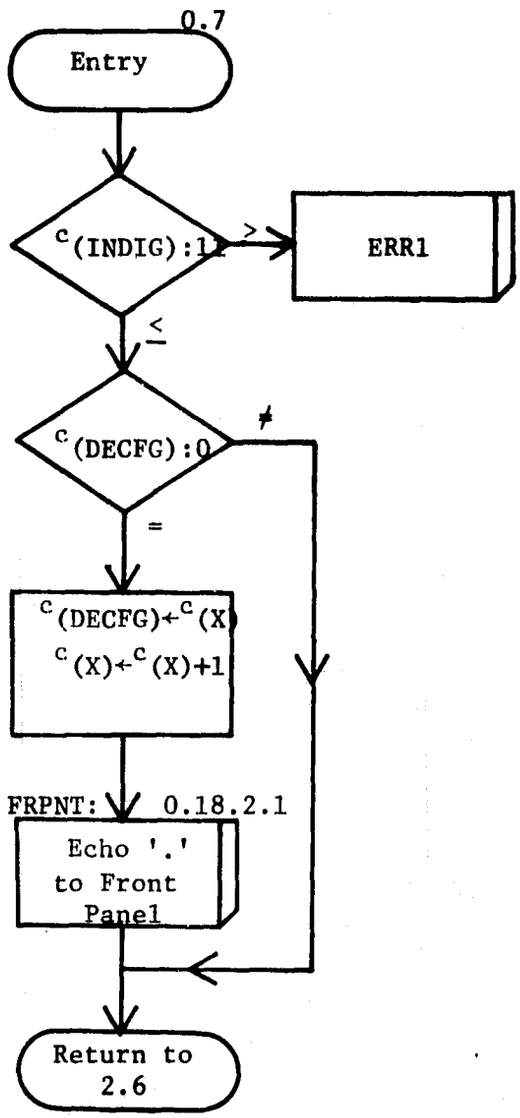
INTERRUPT PROCESSOR, Interrupt-0

RDKEY; Created by a digit key on Front Panel



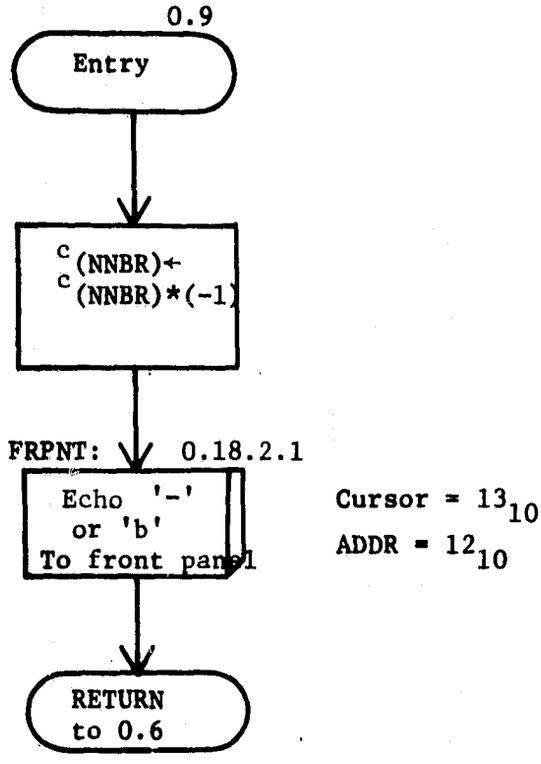
INTERRUPT PROCESSOR, Interrupt-0

DECIM; Digit Key Value ≥ 11



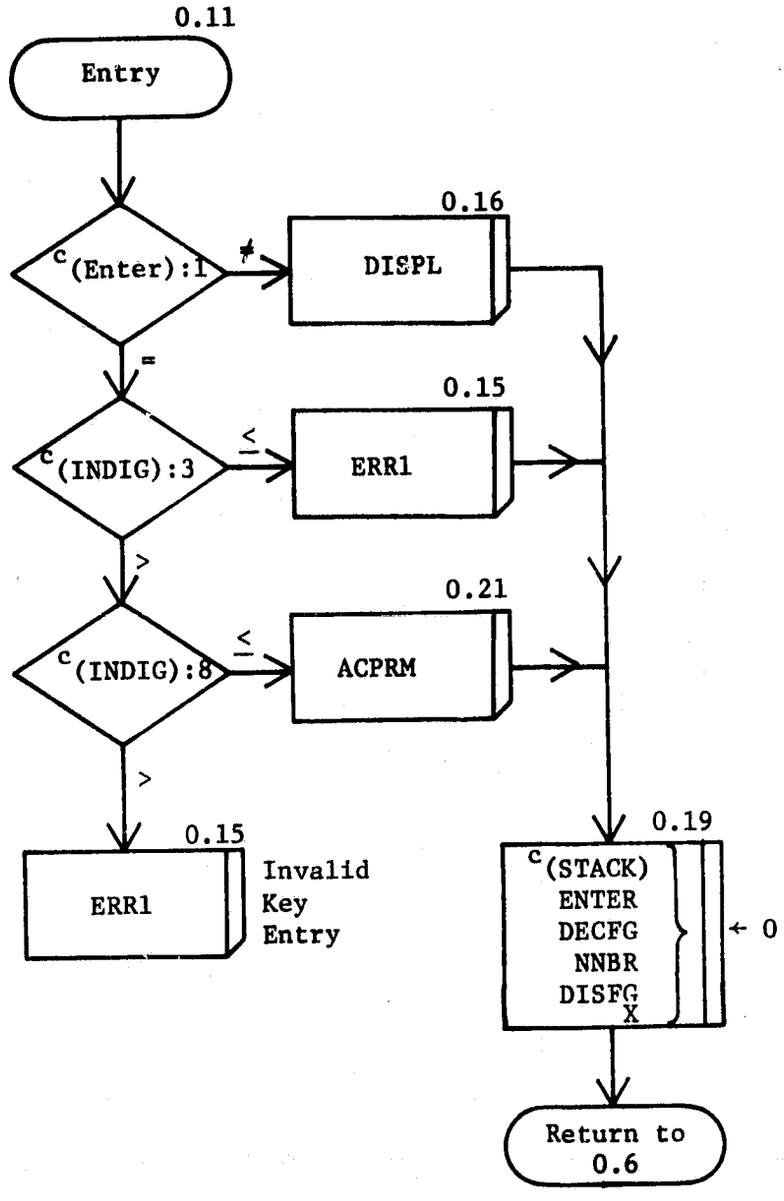
INTERRUPT PROCESSOR, Interrupt-0

CHS; Digit Key Value=10



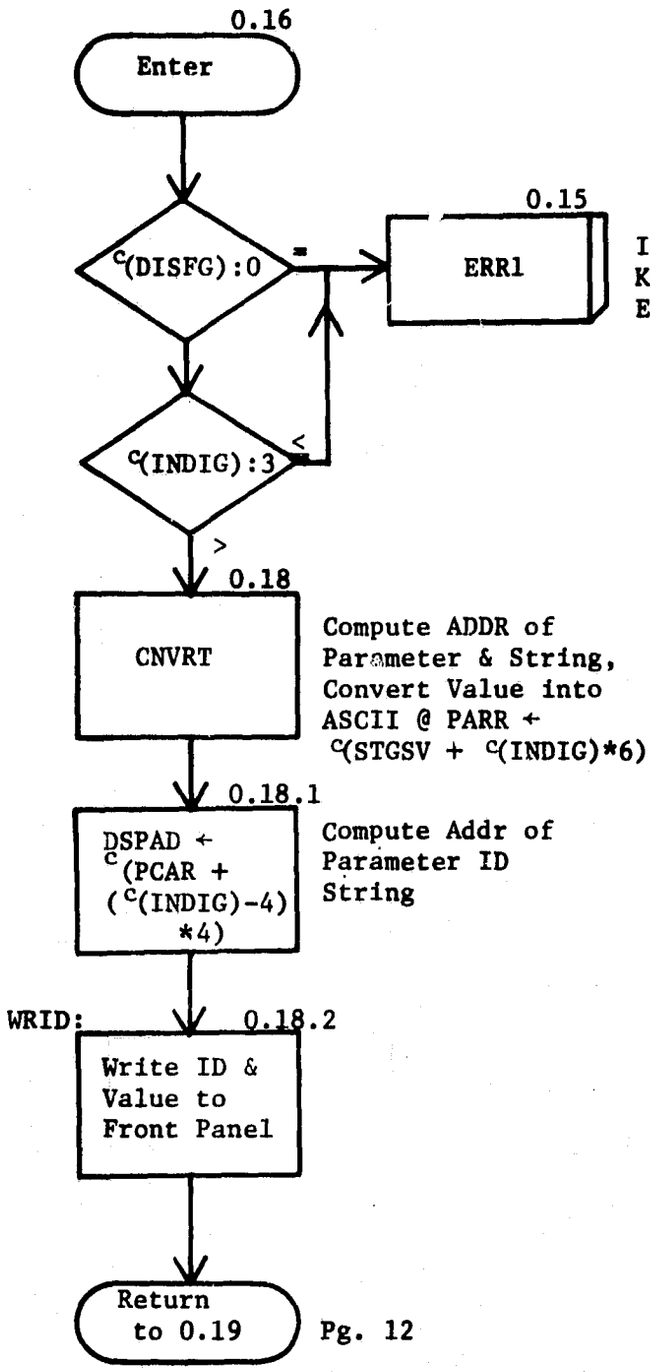
INTERRUPT PROCESSOR, Interrupt-0

PARM. ^{c}(Enter) ≠ 0 & Digit Key Value ≤ 9



INTERRUPT PROCESSOR, PARM Module

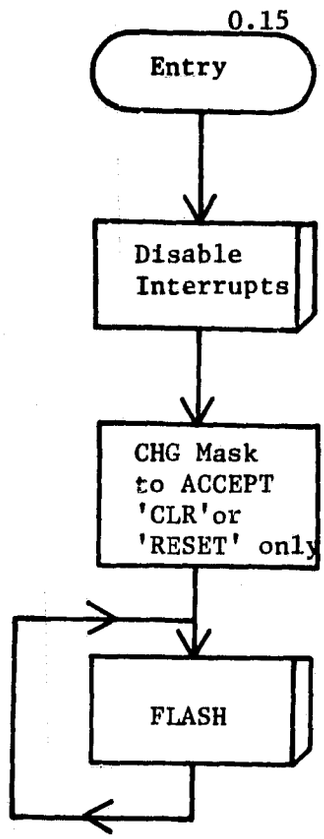
DISPL: Digit Key Value ≤ 9, c(ENTER) ≠ 0, c(ENTER) ≠ 1




```
PCAR:  'DRF  =' ; Character
       'ROL  =' ; Strings for
       'PTC  =' ; Front Panel
       'VEL  =' ; ID Field
       'ALT  =' ; Displays
```

INTERRUPT PROCESSOR, PARM Module

ERR1; Digit Key Value $\leq$ 9, ^{C}(ENTER)=0, ^{C}(X)$\geq$5

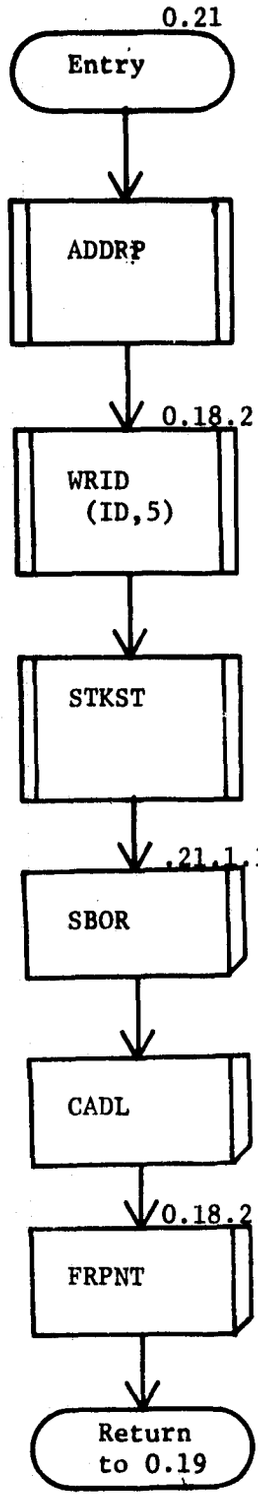


Reset Interrupt Controller to Respond only to 'CLR' function

Flash Front Panel @ 500 ms intervals until 'CLR' or 'RESET' Keyed

INTERRUPT PROCESSOR, PARM Module

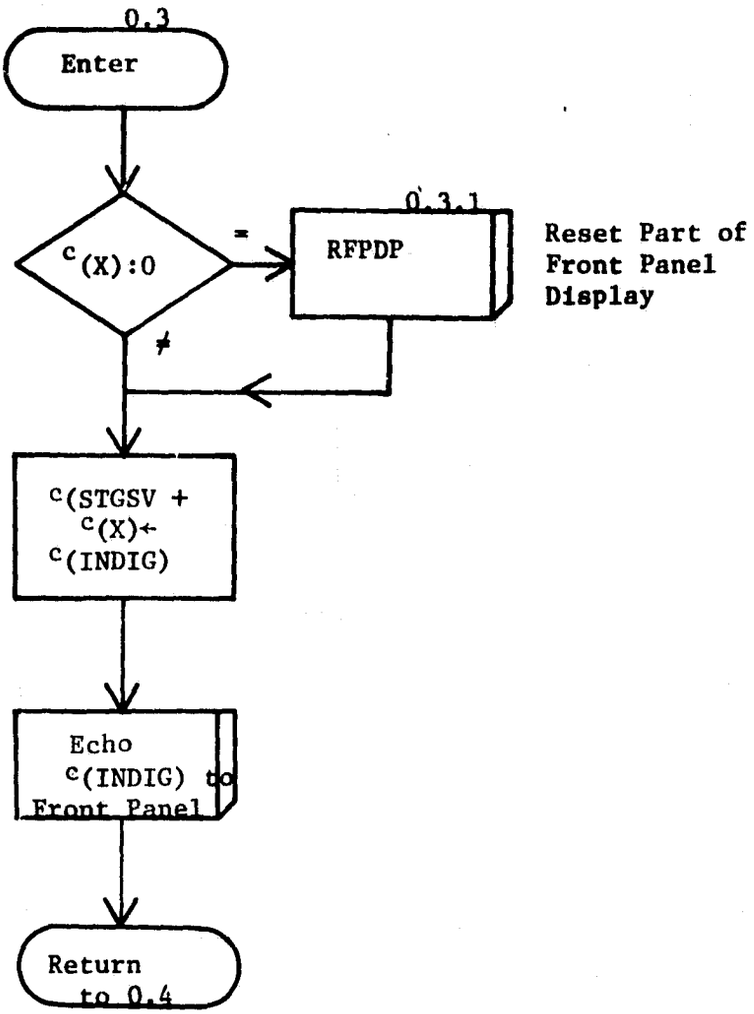
ACPRM; ^{C}(ENTER)≠0, ^{C}(ENTER)=1, 3< Digit Key Value ≤ 8




```
PARV:  0    ; Values Store
       0    ; In FLTG Point
       :
       0    ; Form
STGSV: 0    ; Character String
       0    ; Saved from Front
       :
       0    ; Panel Entries
PCAR: 'DRF ='; Character
      'ROL ='; String for
            . ; F/P ID
            : ; Field
      'ALT=' ;
```

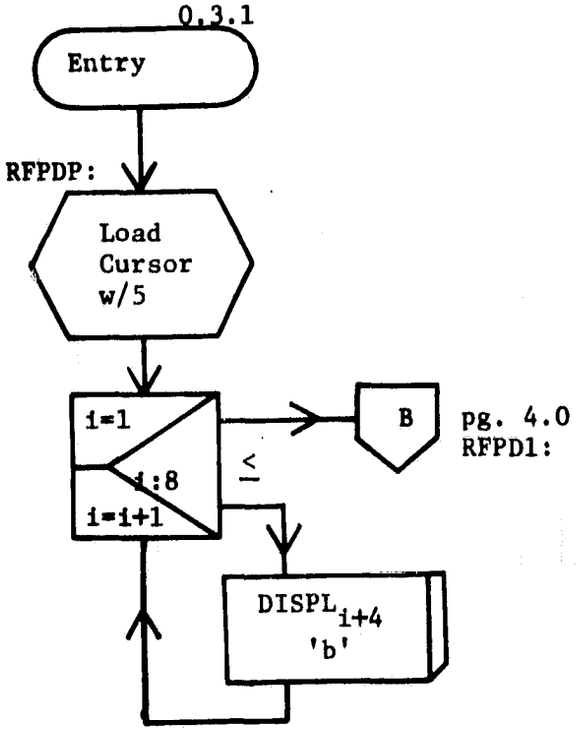
INTERRUPT PROCESSOR, Interrupt-0

DGSTK; Digit Key Value $\leq$ 9, ^{C}(ENTER)=0, ^{C}(X)< 5



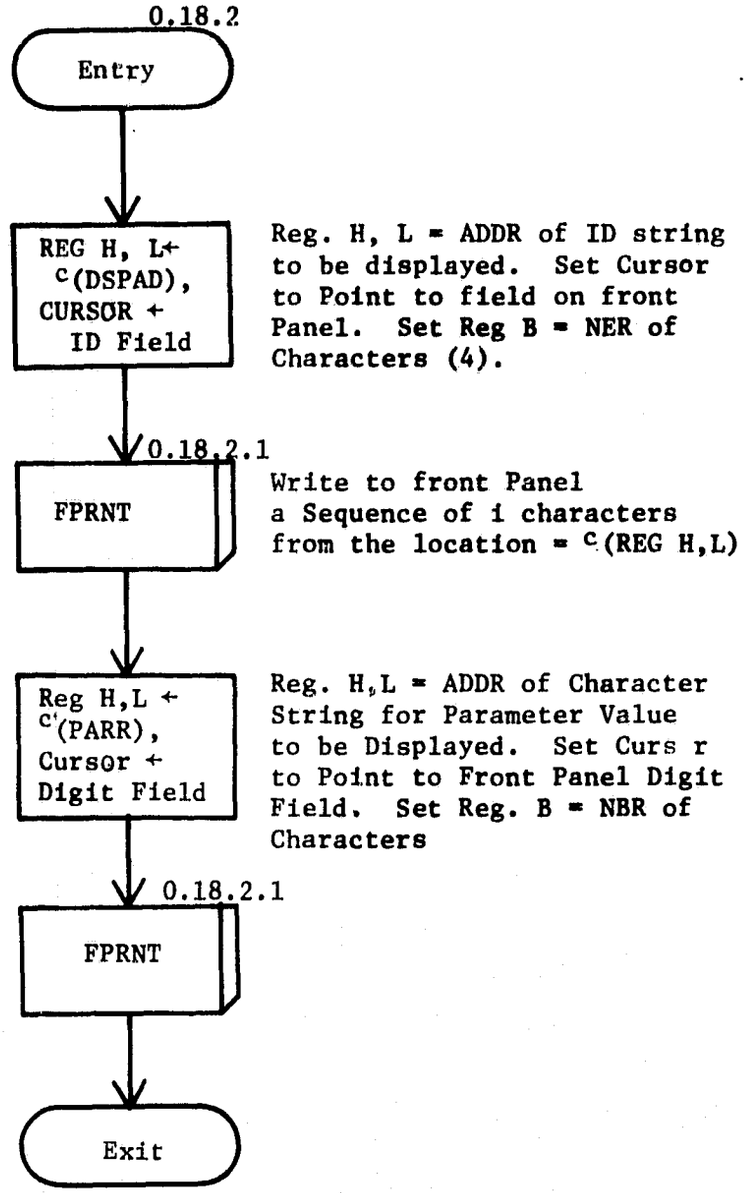
INTERRUPT PROCESSOR,

RFPDP; Reset Part of Front Panel, Interrupt-0, ^{C}(INDIG)<10,
^{C}(ENTER)=0, ^{C}(X)=0



INTERRUPT PROCESSOR, DISPL Module

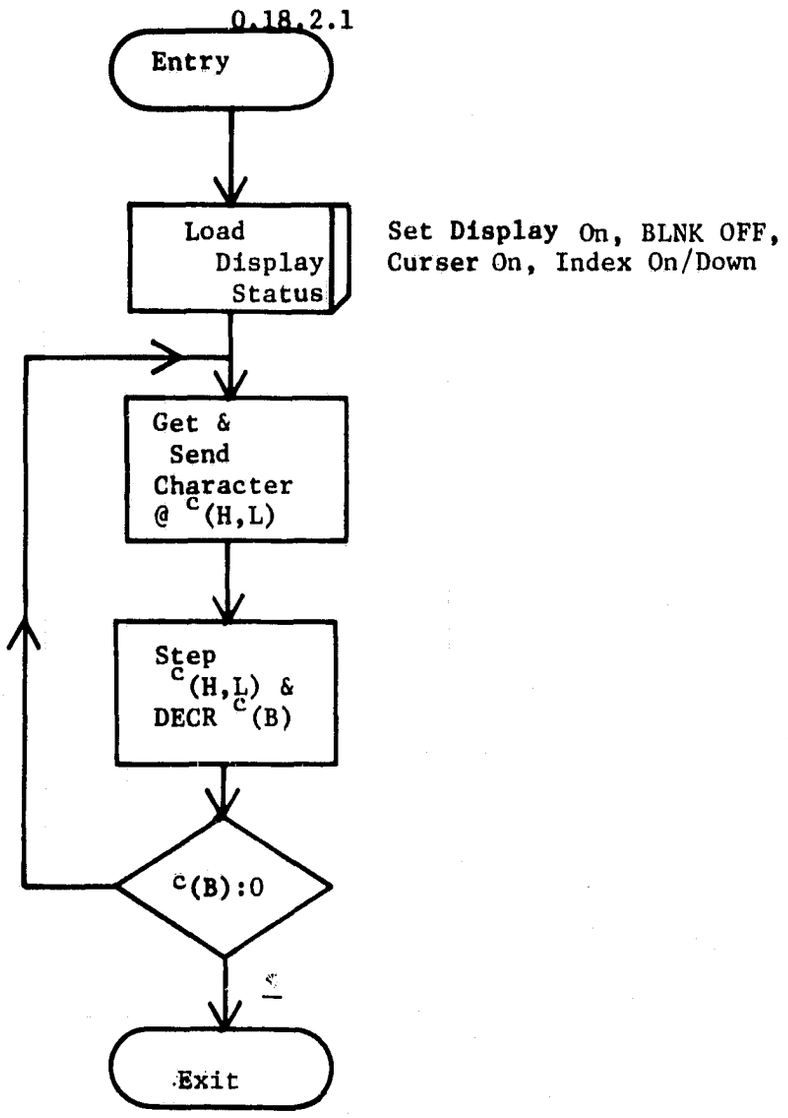
WRID; Write ID to Front Panel, Interrupt-0, c(ENTER) ≠ 0, ^{c}(ENTER) ≠1, ^{c}(DISFG) ≠ 0, ^{c}(INDIG) ≤ 9



INTERRUPT PROCESSOR, DISPL Module, WRID Module

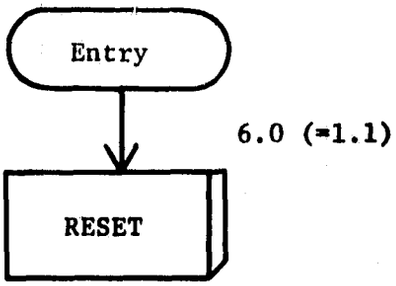
FRPNT, Output Character String to Front Panel,

Reg J,L = ADDR of ASCII String to Display
Reg B = NBR of Characters



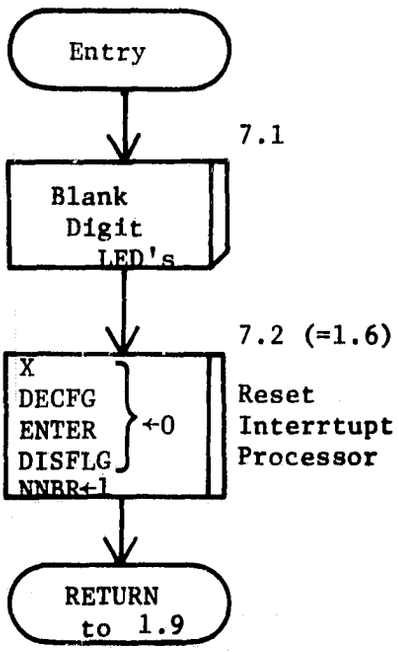
INTERRUPT PROCESSOR

INTERRUPT-6, Reset Key Response



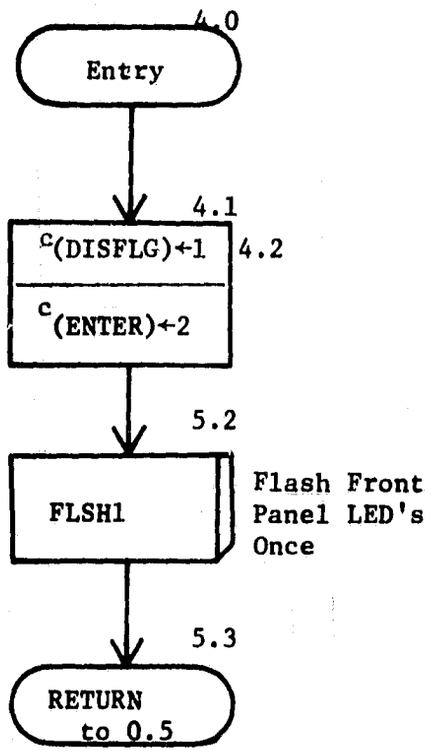
INTERRUPT PROCESSOR

Interrupt-7, CLEAR Key Response.



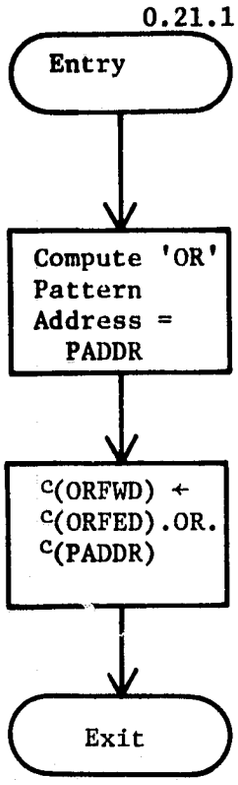
INTERRUPT PROCESSOR

DISPLAY, INTERRUPT-4, Function Key 'DISPL' Depressed



INTERRUPT PROCESSOR, PARM Module, ACPRM Module

SBOR; Set Bit of Over-ride Word, Interrupt-C, c(ENTER) = 1,
3 < Digit Key Value ≤ 8



'OR' Pattern Storage:

ORFWD: 00H; Over-ride Flag Word
ORWRD: 20H; Drift Mask
10H; Roll Mask
08H; Pitch Mask
04H; Vel Mask
02H; ALT Mask

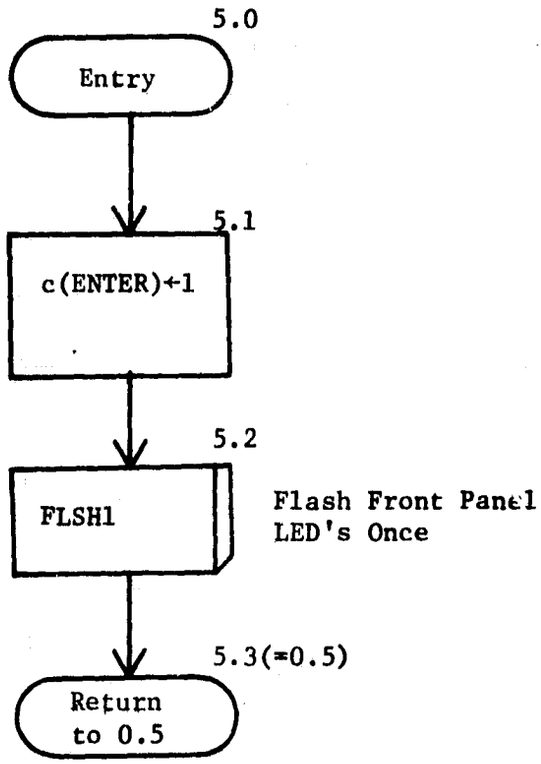
PADDR = ORWRD+c(INDIG)-4
Bit Position=BPOS=^{L}c(INDIG)-1

Over-ride Flag Word Bit Assignments

(MSB) 0	1	2	3	4	5	6	7
(Not used)	(Not used)	Drift	Roll	Pitch	VELOC	Alt	(Not used)

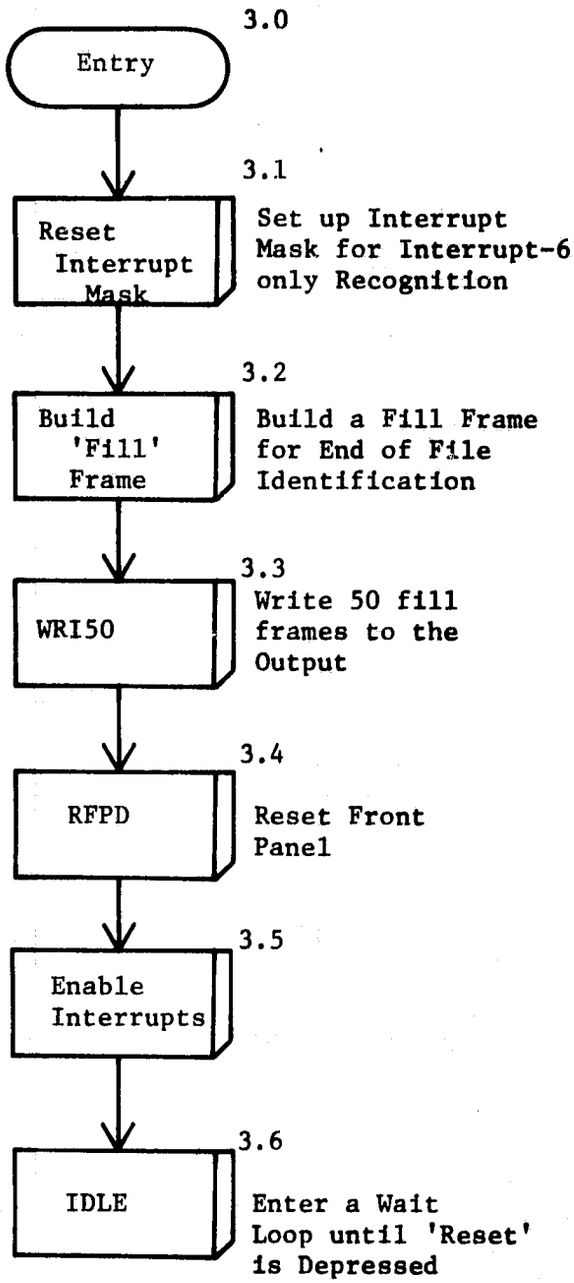
INTERRUPT PROCESSOR

ENTR, INTERRUPT-5, Function Key, 'ENTER', Depressed



INTERRUPT PROCESSOR

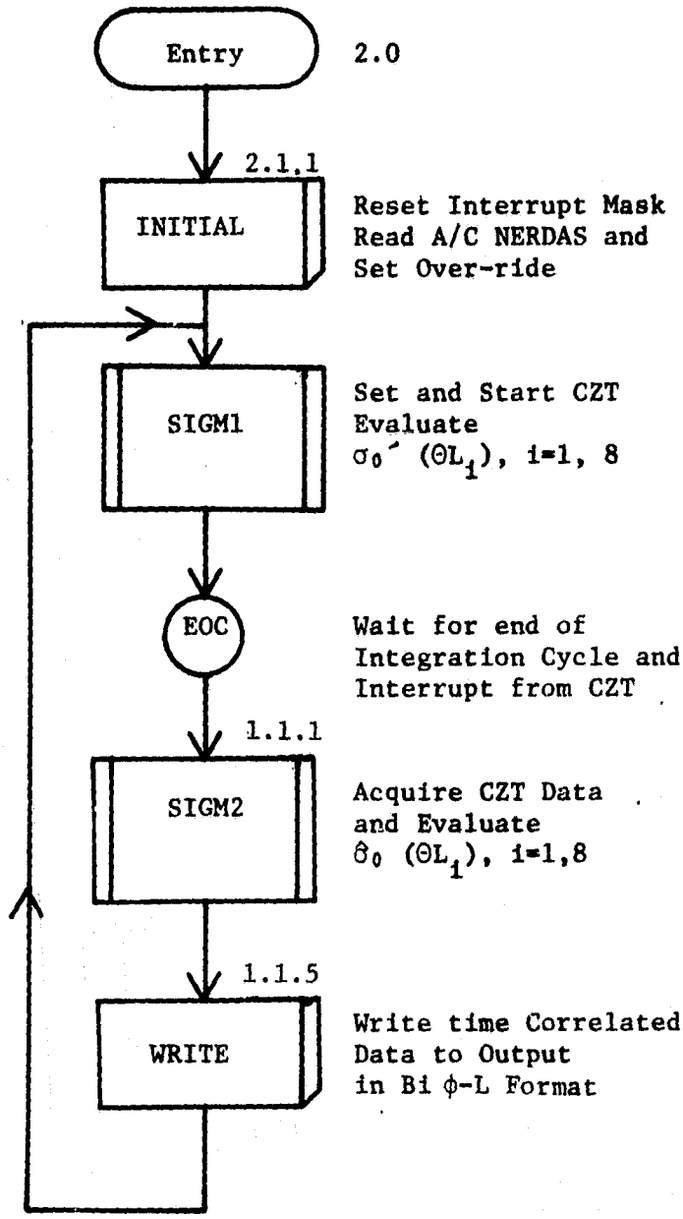
HALT, Interrupt-3, Front Panel State Key, 'HALT', Deressed



RUN MODULE

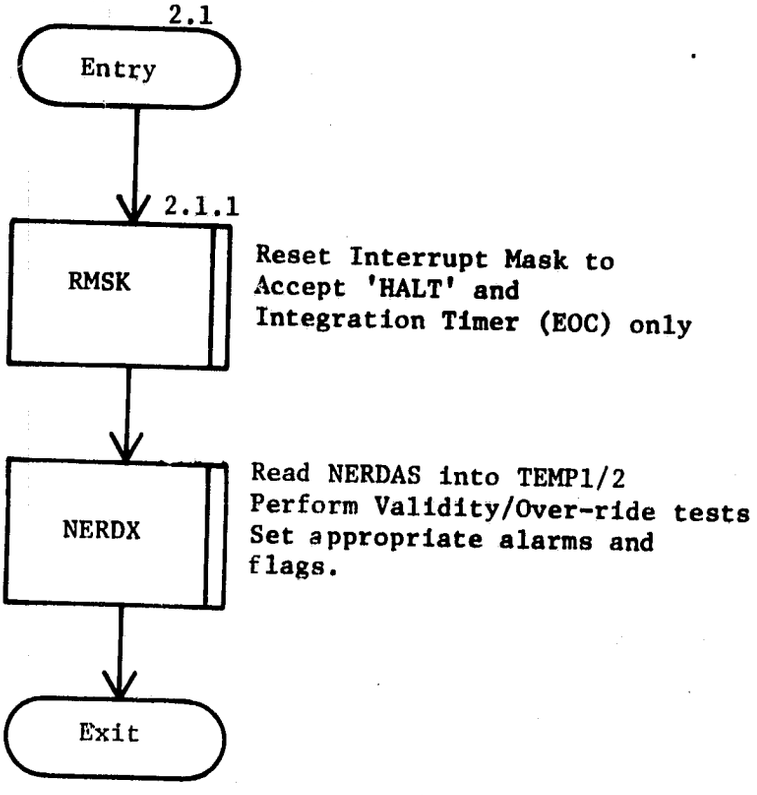
INTERRUPT-2, Key Labelled 'RUN'

RUN Module



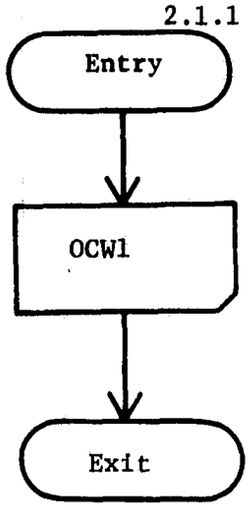
INTERRUPT-2 PROCESSOR, RUN Module

INITAL, Key Labelled 'RUN' depressed.



INTERRUPT-2 PROCESSOR, RUN Module, INITIAL Module

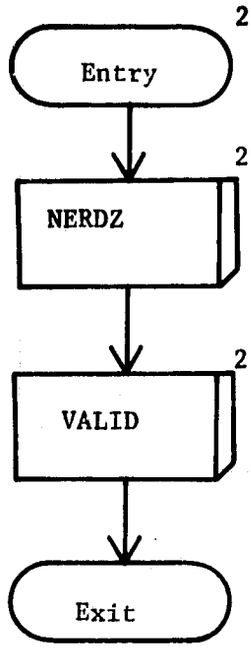
RMSK; Run Mask Set, Interrupt-2, 'RUN' Key Depressed



Issue OCWi to the 8259, Location
00DBH, for Mask;
OCW1 = 00F5H, Masking all but
'HALT' and 'EOC' Interrupts

INTERRUPT-2 PROCESSOR, Run Module, Inital Module

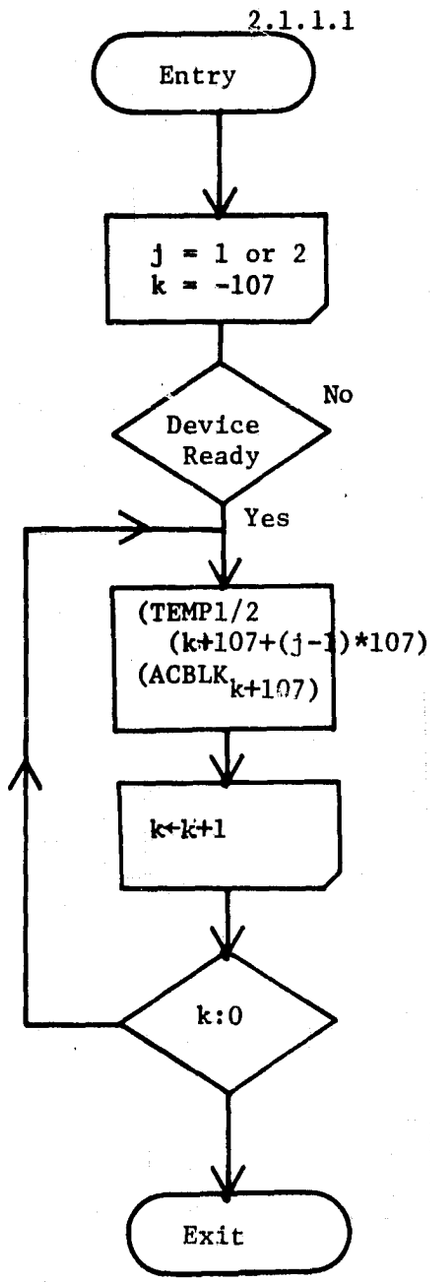
NERDX: Interrupt-2, SIGM1 Module
Reg(H,L) = Destination Starting Address



INTERRUPT-2 PROCESSOR, RUN Module, INITIAL Module, NERD1 Module

NERDZ: NERDX Routine, SIGM1 Module, Interrupt-2

Decode & Block A/C Data into TEMP1 or TEMP2 Buffer



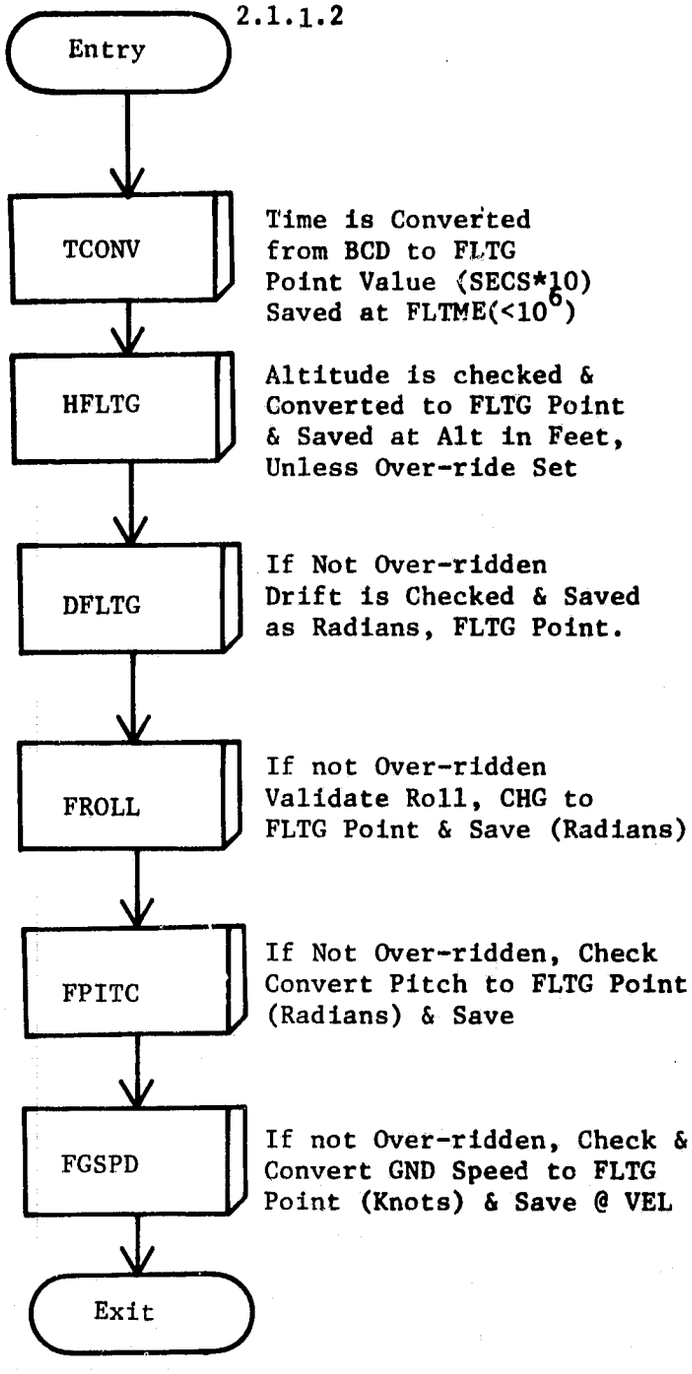
Storage Arrangement:

```
TEMP1:  0 ; BCD Values-of
        0 ; A/C Data, Each
        0 ; WORD = 4 LSB's
        .
        .
        .
        0 ; Of 107 WORDS
        0 ; Of ACBLK, Thru
        0 ; Run Number

TEMP2:  0 ; BCD Value of
        0 ; A/C Data, each
        0 ; WORD=4LSB's
        .
        .
        .
        0
        0
        0 ;
```

INTERRUPT-2 PROCESSOR, RUN Module, INITIAL Module, NERD1 Module

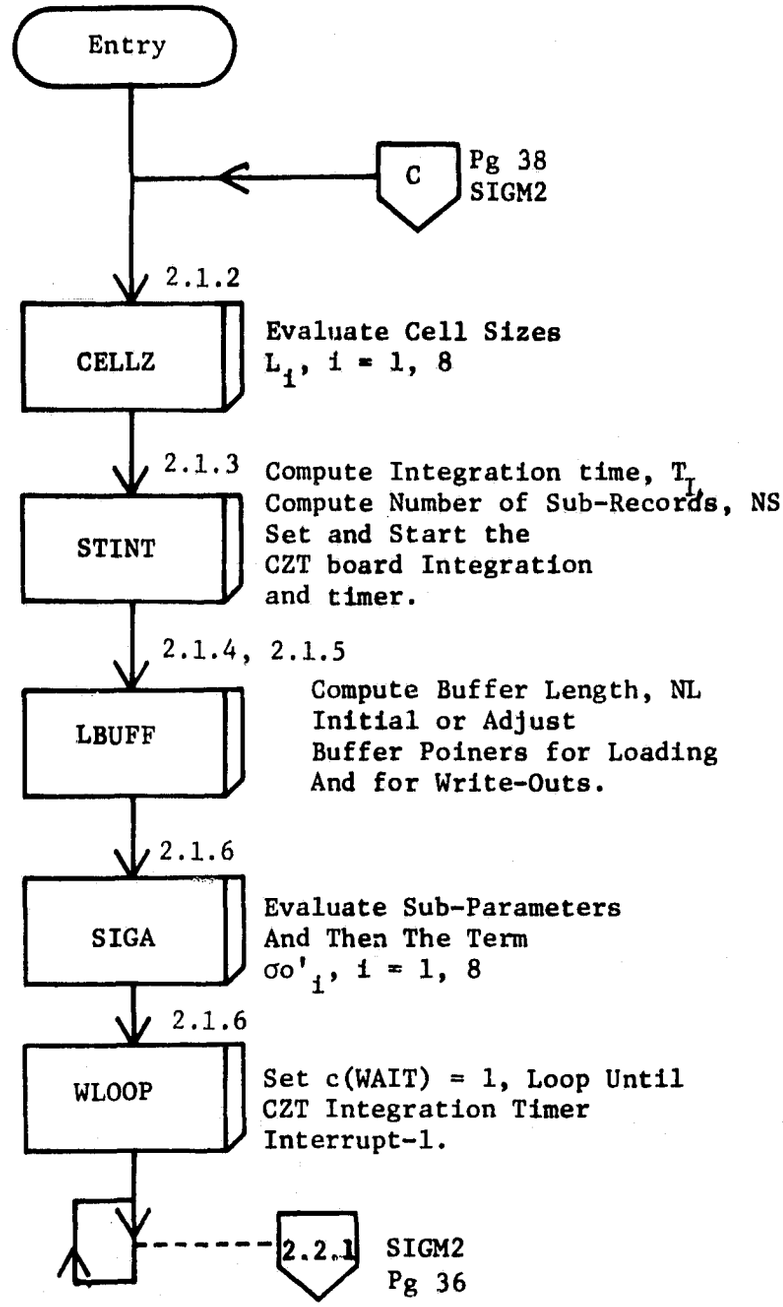
VALID: NERD1 Routine, SIGM1 Module, Interrupt-2
Converts Time & Other A/C Parameters
to FLTG Point. Where Appropriate, Over-ride Values are
Substituted & Alarms & Flags are Set.




```
FLTME:  ;Cell Time
ALT:    ;Altitude
DRFT:   ;Drift (RAD)
ROLL:   ;Roll (RAD)
PITCH:  ;Pitch (RAD)
  VEL:  ;GND Speed
```

INTERRUPT-2 PROCESSOR, RUN Module

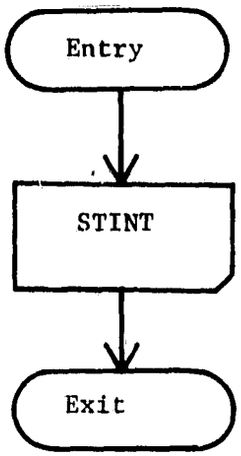
SIGM1; Key Labelled 'RUN' depressed.



INTERRUPT-2 PROCESSOR, Run Module, SIGM1 Module

STINT; Start Integration

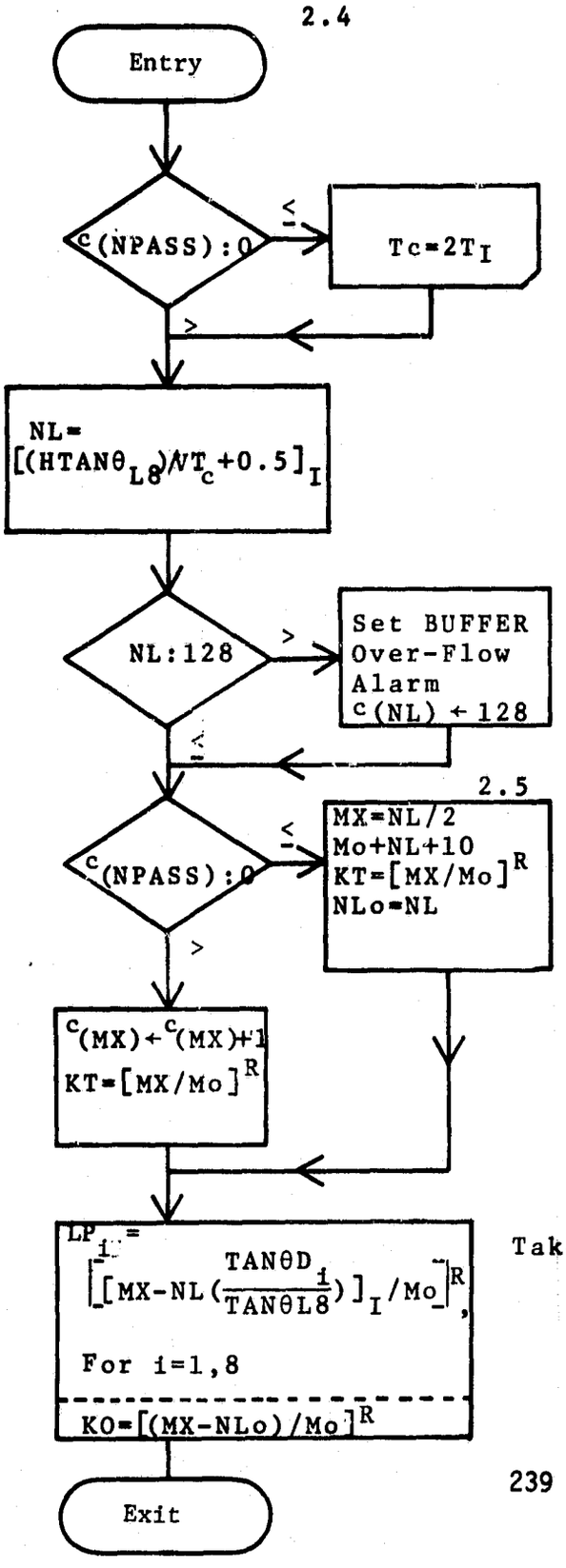
2.1.3



Evaluate
$NS=[(Tk/512)fs+0.5]_I$

Send Value of NS to CZT Board, starting Integration

INTERRUPT-2 PROCESSOR, RUN Module, SIGMA1 Module
LBUFF Submodule, BUFFER Load Set-Up

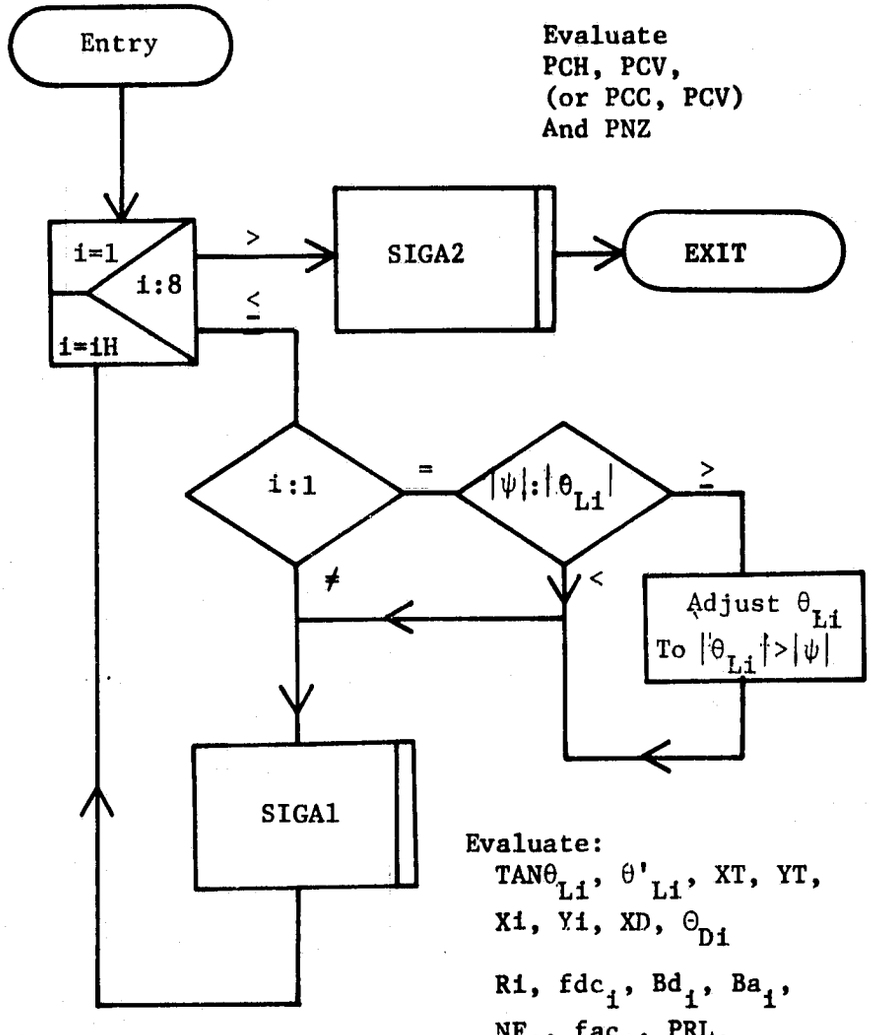


Take Care of LP<0

INTERRUPT-2 PROCESSOR, RUN MODULE, SIGM1 Module

SIGA; $\sigma o'_i$ evaluation

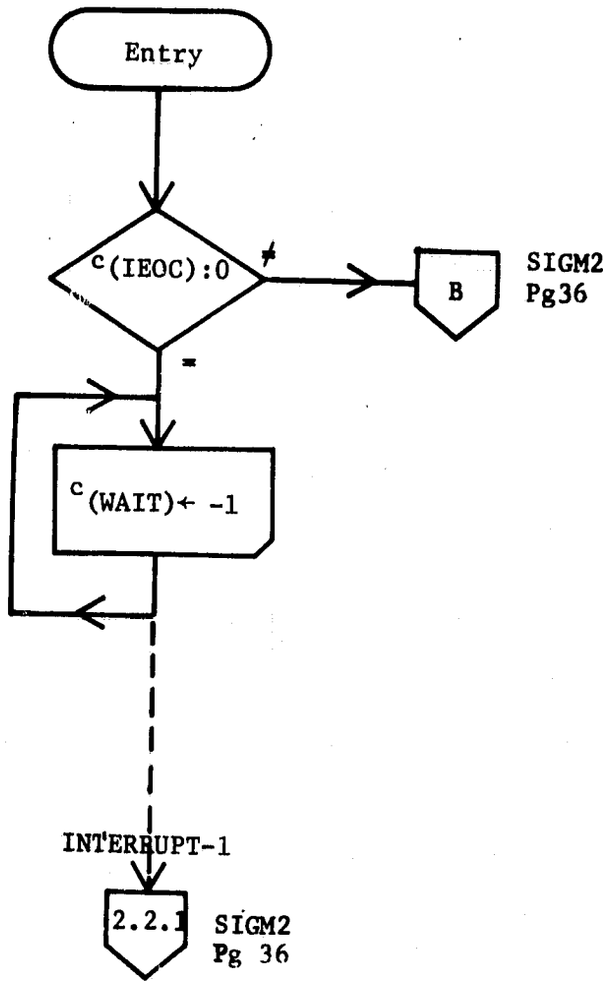
2.1.6



INTERRUPT-2 PROCESSOR, RUN Module, SIGM1 Module

WLOOP; Wait Loop.

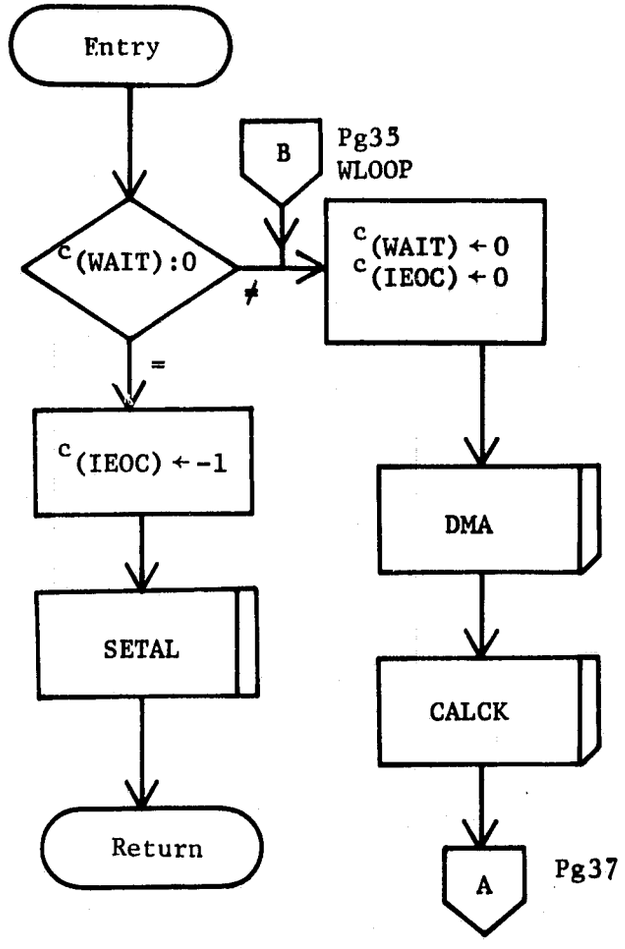
2.1.7



INTERRUPT-1 PROCESSOR, RUN MODULE

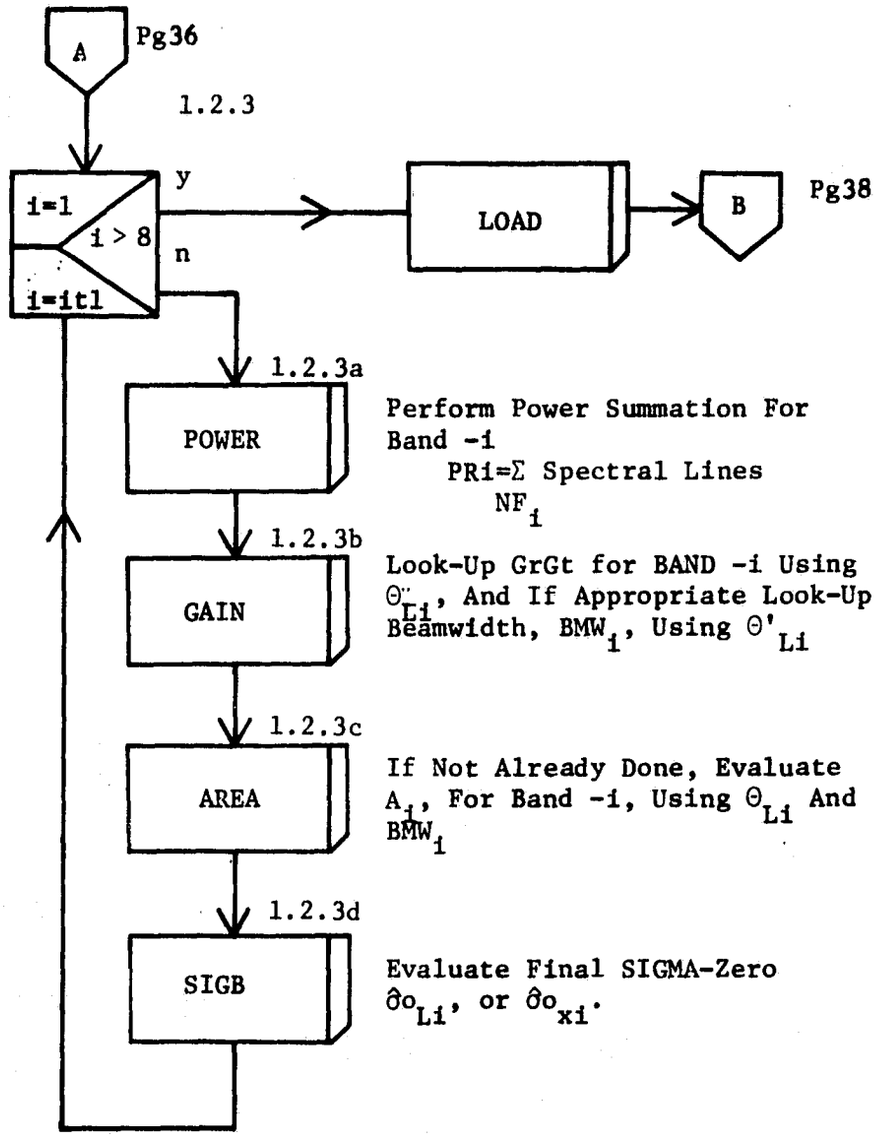
SIGM2; End of Integration Time (EOG), Or
Exit From WLOOP With IEOC Set.

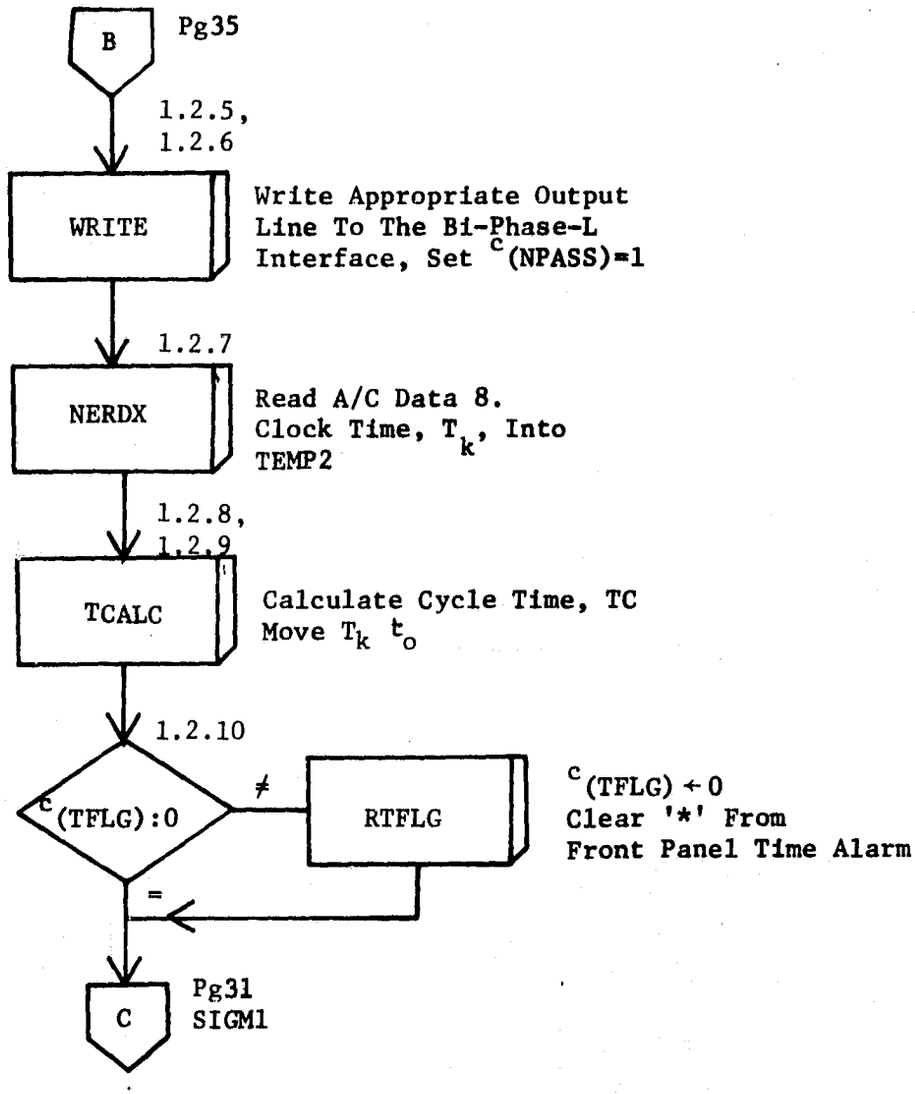
1.2.1



INTERRUPT-1 PROCESSOR, RUN Module

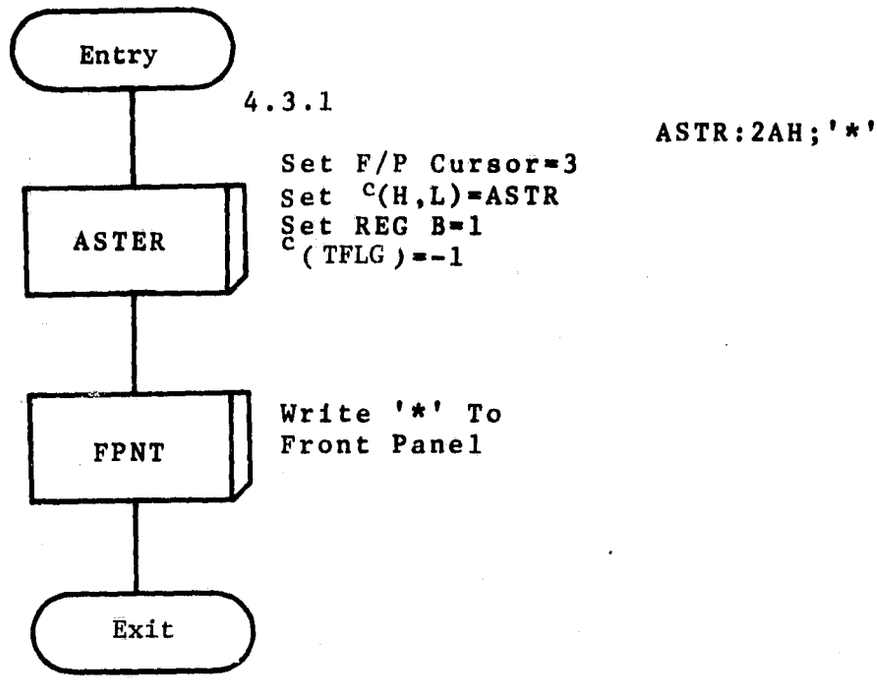
SIGM2: (Cont. A)




B
Pg35
1.2.5,
1.2.6
WRITE
Write Appropriate Output
Line To The Bi-Phase-L
Interface, Set C(NPASS)=1
1.2.7
NERDX
Read A/C Data 8.
Clock Time, T_k, Into
TEMP2
1.2.8,
1.2.9
TCALC
Calculate Cycle Time, TC
Move T_k t_o
1.2.10
C(TFLG):0
≠
RTFLG
C(TFLG) ← 0
Clear '*' From
Front Panel Time Alarm
=
C
Pg31
SIGM1

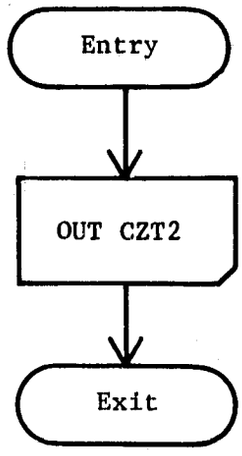
INTERRUPT-1 PROCESSOR, RUN Module, SIGM2 Module

SETAL; SETS ALARM FLAGS ON FRONT PANEL



INTERRUPT-1 PROCESSOR, RUN Module, SIGM2 Module

DMA; Initiates DMA Transfer of CZT FREQ Components

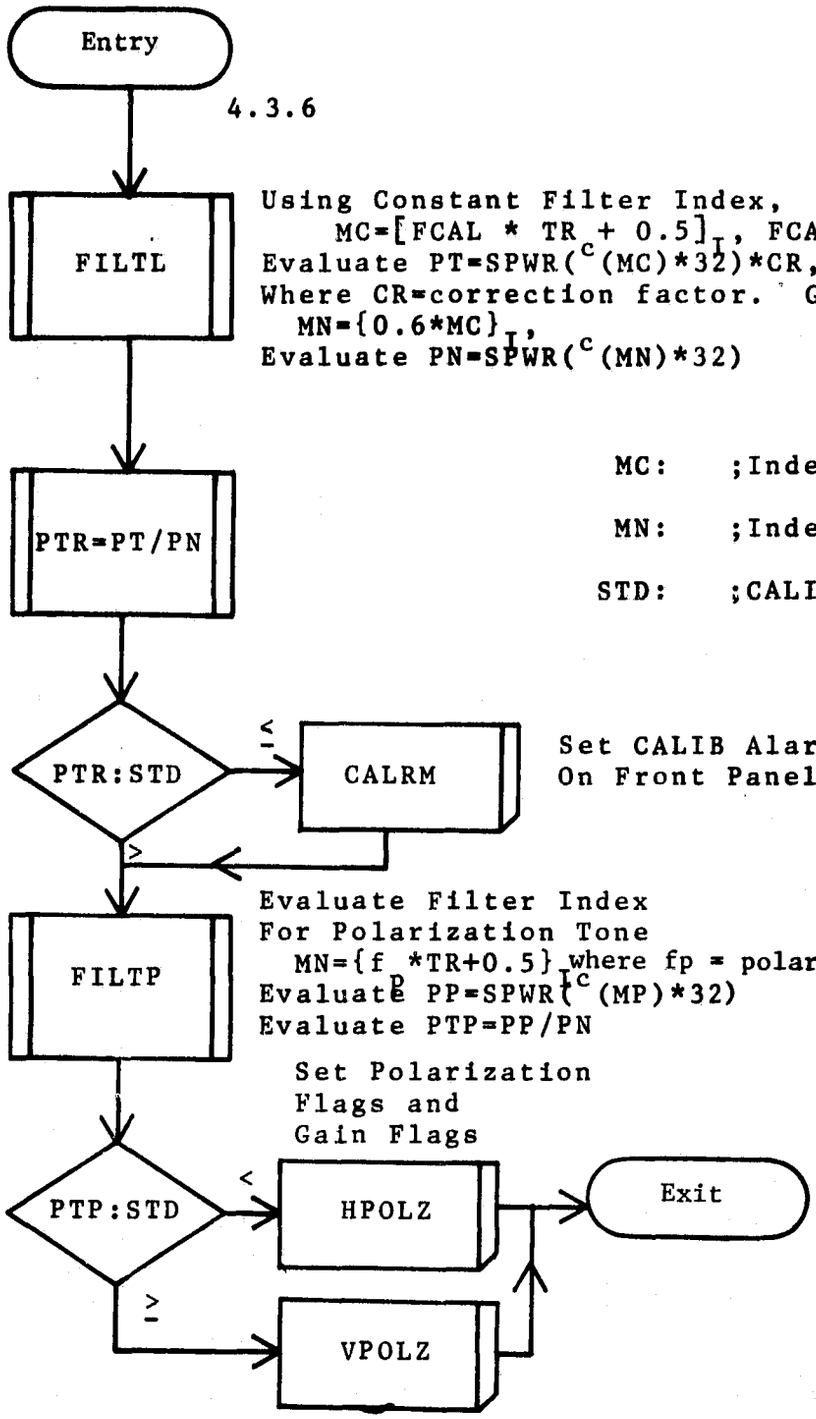



```
CZT1  EQU 08H
CZT2  EQU 09H
CZT3  EQU 0AH
CZT4  EQU 0BH
```

```
SPWR:0    ;512 x 32
     0    ;BIT
     0    ;BUFFER FOR
     :    :
     .    .
     0    ;SPECTRAL
     0    ;POWER
```

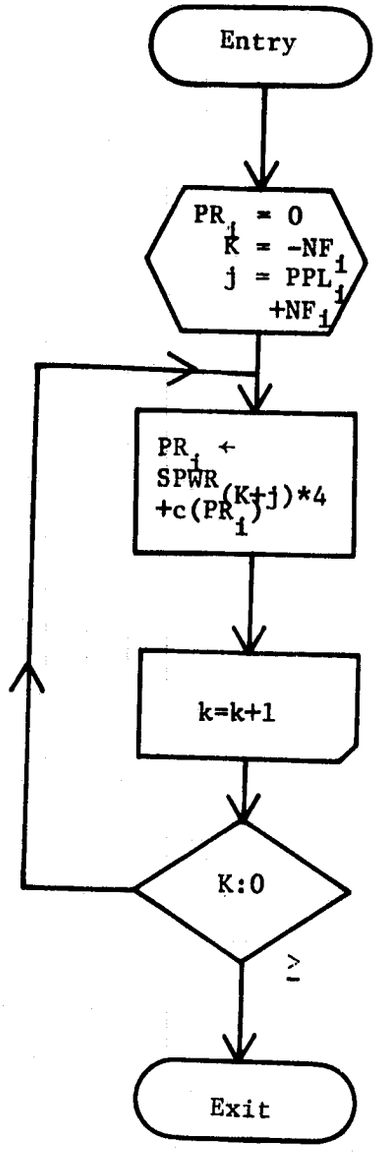
INTERRUPT-1 PROCESSOR, RUN Module, SIGM2 Module

CALCK; Evaluate Power Transmitted and Decodes Polarization (For L-Band System)



INTERRUPT-1 PROCESSOR, RUN Module, SIGM2 Module

POWER; Computer Spectral Power for Each Doppler Band

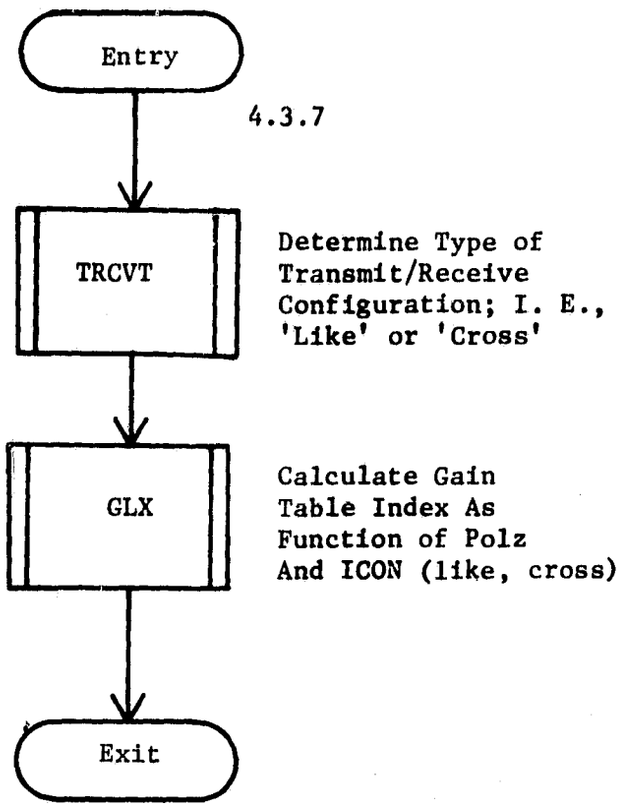



```
SPWR:0    ;Spectral
      0   ;Power
      :   ;Buffer
      0
      0   ;512x4 Words
```

```
PR:0      ;Received Power
      0   ;For Each
      0   ;Doppler Band
      :
      0   ;Up to 8 Values
```

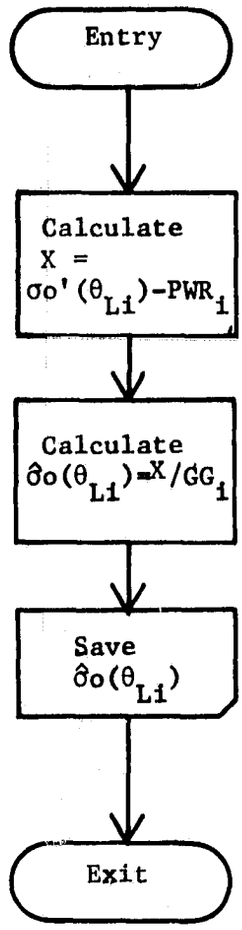
INTERRUPT-1 PROCESSOR, RUN Module, SIGM2 Module

GAIN; Determine Gain Values From Table (L-Band)



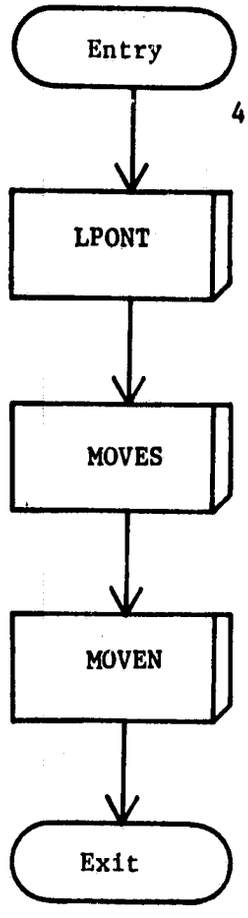
INTERRUPT-1 PROCESSOR, RUN Module, SIGM2 Module

SIGB; Computes $\hat{\sigma o}_{Li}$ or $\hat{\sigma o}_{xi}$ For Each Doppler Band



LOAD; SIGM2 ROUTINE, INTERRUPT-1

Computes Pointers into Out-Buffer, Moves Nerdas, $\hat{\sigma}o(\alpha_k)$ Into Appropriate Lines And Columns. Moves Balances of Other Out-Put Parameters Into Out-Buffer.



Evaluate Line Pointer For NADIR & For Each Look Angle

$LP_k = (DFA_k/256)*N, k=1, 8$

$LINE_k = MOD((LDPTR-LP_k),(N+10)), k=1, 8$

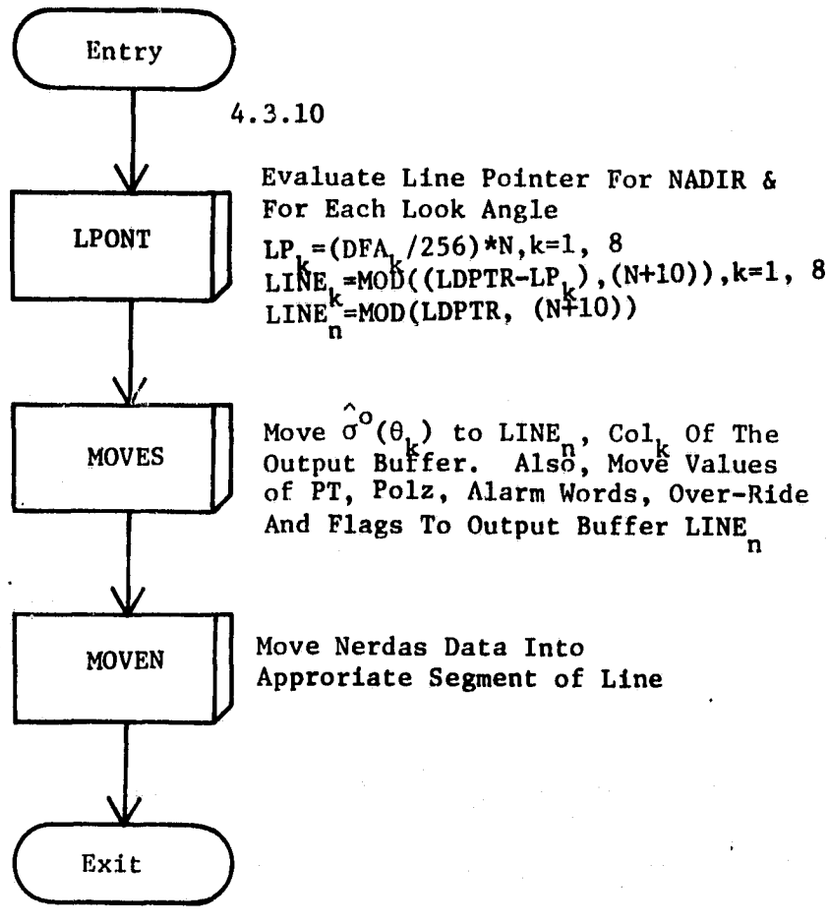
$LINE_n^k = MOD(LDPTR, (N+10))$

Move $\hat{\sigma}^o(\theta_k)$ to $LINE_n$, Col_k Of The Output Buffer. Also, Move Values of PT, Polz, Alarm Words, Over-Ride And Flags To Output Buffer $LINE_n$

Move Nerdas Data Into Approriate Segment of Line

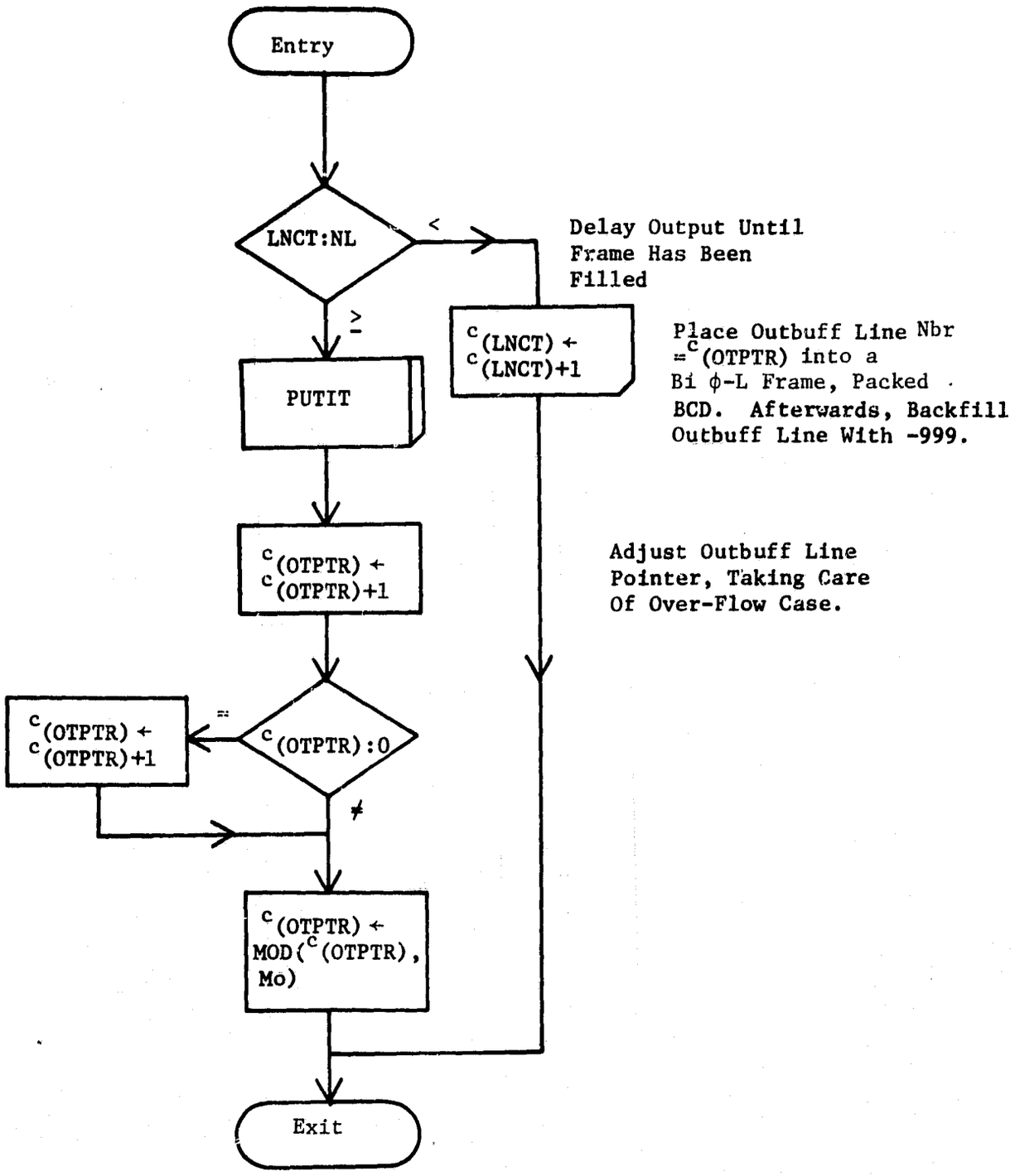
LOAD; SIGM2 ROUTINE, INTERRUPT-1

Computes Pointers into Out-Buffer, Moves Nerdas, $\hat{\sigma}o(\alpha_k)$ Into Appropriate Lines And Columns. Moves Balances of Other Out-Put Parameters Into Out-Buffer.



INTERRUPT-1 PROCESSOR, RUN Module, SIGM2 Module

WRITE;



OUTPUT BUFFER LINE CONTENTS

ADDRESS	BYTE (BCD) CONTENTS		ADDRESS	BYTE (BCD) CONTENTS	
XX 00	*0.1		XX 13	*0.1	
01	*1	SEC	14	*1	LONG, MIN
02	*10		15	*10	
03	*1	MIN	16	*1	
04	*10		17	*10	LONG, DEG
05	*1	HRS	18	*100	
06	*10		19	SIGN	
07	*1	DAY	1A	*1	
08	*10		1B	*10	
09	*1	MONTH	1C	*100	ALT
0A	*10		1D	*1000	
0B	*1	YR	1E	*10000	
0C	*10		1F	*0.1	
0D	*0.1		20	*1	HDG
0E	*1	LAT, MIN	21	*10	
0F	*10		22	*100	
10	*1		23	*0.1	
11	*10	LAT, DEG	24	*1	DRIFT
12	SIGN		25	*10	
			26	SIGN	

OUTPUT BUFFER LINE CONTENTS (CONT.)

ADDRESS	BYTE (BCD) CONTENTS		ADDRESS	BYTE (BCD) CONTENTS	
XX 27	*0.1		XX 3C	*0.1	
28	*1	ROLL	3D	*1	THETA-A(1)
29	*10		3E	*10	
2A	SIGN		3F	*0.1	
2B	*0.1		40	*1	THETA-A(2)
2C	*1	PITCH	41	*10	
2D	*10		42	*0.1	
2E	SIGN		43	*1	TAETA-A(3)
2F	*1		44	*10	
30	*10	GND SPEED	45	*0.1	
31	*100		46	*1	THETA-A(4)
32	*1		47	*10	
33	*10	MISSION	48	*0.1	
34	*100		49	*1	THETA-A(5)
35	*1		4A	*10	
36	*10	SEQ	48	*0.1	
37	*100		4C	*1	THETA-A(6)
38	*1	FLT	4D	*10	
39	*10		4E	*0.1	
3A	*1	LINE	4F	*1	THETA-A(7)
38	*10		50	*10	

OUTPUT BUFFER LINE CONTENTS (CONT.)

ADDRESS	BYTE (BCD) CONTENTS	
XX 51	*0.1	
52	*1	THETA-A(8)
53	*10	
54	*0.1	
55	*1	SIGMA(1)
56	*10	
57	SIGN	
58	*0.1	
59	*1	SIGMA(2)
5A	*10	
5B	SIGN	
5C	*0.1	
5D	*1	SIGMA(3)
5E	*10	
5F	SIGN	
60	*0.1	
61	*1	SIGMA(4)
62	*10	
63	SIGN	

ADDRESS	BYTE (BCD) CONTENTS	
XX 64	*0.1	
65	*1	SIGMA(5)
66	*10	
67	SIGN	
68	*0.1	SIGMA(6)
69	*1	
6A	*10	
6B	SIGN	
6C	*0.1	
6D	*1	SIGMA(7)
6E	*10	
6F	SIGN	
70	*0.1	
71	*1	SIGMA(8)
72	*10	
73	SIGN	
74	*0.1	
75	*1	PT (CALIB. PWR)
76	*10	
77	SIGN	

OUTPUT BUFFER LINE CONTENTS (CONT.)

ADDRESS	BYTE (BCD) CONTENTS	
XX 78		Polarization (HH-11H, HV=10H, VV=00H, VH=01H)
79		BUF Alarm (0=OFF, 1=ON)
7A		Time Alarm (0=OFF, 1=ON)
7B		Calib. Alarm (0=OFF, 1=ON)
7C		Over-Ride Bits
7D		Last-Value Bits
7E		FFH
7F		FFH End of Frame

APPENDIX G

APPENDIX G

INTEL SBC 80/20 SOFTWARE

```
8080 MACRO ASSEMBLER, VER 2.0          ERRORS = 0-PAGE 1

;
;**************************************************
;
;  REMOTE SENSING CENTER
;  TEXAS A&M UNIVERSITY
;
;  PROJECT 3556
;
;**************************************************
;
;  DESCRIPTION
;  ===========
;
;  THIS PROGRAM CONTAINS ALL SOFTWARE DEVELOPED
;  FOR THE INTEL SBC 80/20 DURING PROJECT 3556.
;  THERE ARE 5 PROGRAMS WITHIN THE MAIN
;  PROGRAM.  UPON EXECUTION OF THE PROGRAM,
;  A MENU IS PRINTED.  THE USER MAY SELECT THE
;  DESIRED PROGRAM FROM THE MENU.  THE SELECTED
;  PROGRAM WILL EXECUTE AUTOMATICALLY, PROMPTING
;  THE OPERATOR FOR ANY ADDITIONAL REQUIRED
;  INFORMATION.
;
;  THE 5 PROGRAMS ARE:
;
;  1)  'SAMPLE ACCUMULATION AND DMA TRANSFER'
;      THIS PROGRAM INITIALIZES THE CZT
;      HARDWARE, PERFORMS  SAMPLE ACCUMULATION,
;      AND DOES THE DMA TRANSFER TO THE SBC
;      80/20 AT MEMORY LOCATION 6000 HEX.  TO
;      TERMINATE THE PROGRAM, HIT 'ESC' WHEN
;      PROMPTED FOR NUMBER OF RECORDS.
;
;  2)  'SAMPLE ACCUMULATION,DMA TRANSFER AND
;      TRANSFER TO TI'
;      THIS PROGRAM PERFORMS EXACTLY AS PROGRAM
;      1, EXCEPT, AT THE END OF THE DMA TRANSFER
;      THE DATA IS TRANSFERED TO THE TI 990 MINI-
;      COMPUTER.  THE PROGRAM 'TX8080' MUST
;      BE RUNNING ON THE TI FOR A TRANSFER TO
;      TAKE PLACE.
;
;  3)  'ADC ALIGN'
;      THIS PROGRAM IS USED TO ALIGN THE ANALOG
;      TO DIGITAL CONVERTERS (ADC).  IT CAUSES
```

```
                        ;       THE PROCESSOR TO TAKE 256 RECORDS, DO
                        ;       A DMA TRANSFER AND IMMEDIATELY START
                        ;       TAKING MORE RECORDS.  SINCE THE DMA
                        ;       REQUIRES ONLY ABOUT 50 MSEC TO COMPLETE
                        ;       IT GIVES THE IMPRESSION THAT THE PROCESSOR
                        ;       IS IN A MODE OF CONTINUOUS SAMPLE
                        ;       ACCUMULATION.  TO TERMINATE THE
                        ;       PROGRAM, PRESS 'ESC'.
                        ;
                        ;  4)   'CASSETTE LOAD'
                        ;       THIS PROGRAM WILL LOAD PROGRAMS OR
                        ;       DATA STORED ON CASSETTE TAPES FROM THE
                        ;       SILENT 700 TERMINAL TO THE S-100 80/20
                        ;       MEMORY.  CASSETTE DRIVE 1 MUST BE USED,
                        ;       AND THE UNIT MUST BE IN THE PLAYBACK
                        ;       MODE.  THE OPERATOR MUST POSITION THE
                        ;       TAPE TO THE BEGINNING OF THE DESIRED
                        ;       PROGRAM.  THE PROGRAM PROMPTS THE OPERATOR
                        ;       FOR THE MEMORY ADDRESS WHERE THE PROGRAM
                        ;       ON TAPE IS TO BE STORED.  UPON COMPLETION
                        ;       CONTROL IS RETURNED TO THE MONITOR.
                        ;
                        ;  5)   'CASSETTE PUNCH'
                        ;       THIS PROGRAM WILL RECORD PROGRAMS FROM
                        ;       COMPUTER MEMORY TO CASSETTE TAPE.
                        ;       CASSETTE DRIVE 1 MUST BE USED, AND THE
                        ;       UNIT MUST BE IN RECORD MODE.  THE
                        ;       OPERATOR IS PROMPTED FOR STARTING AND
                        ;       ENDING ADDRESSES OF DATA OR PROGRAM TO
                        ;       BE RECORDED.  THE 'RECORD' REFERED TO
                        ;       IN THE OPERATOR INSTRUCTIONS IS THE
                        ;       ON/OFF SWITCH IN THE RECORD CONTROL
                        ;       SECTION OF THE SILENT 700.  UPON COM-
                        ;       PLETION, CONTROL IS RETURNED TO THE
                        ;       MONITOR.  THE RECORD CONTROL MUST BE
                        ;       TURNED OFF BEFORE THE TAPE CAN BE MOVED.
                        ;
                        ;
                        ;***************************************************
                        ;
                        ;  PROGRAM EQUATES
                        ;
                        ;***************************************************
                        ;
00FF                    ICW1    EQU   0FFH
0008                    ICW2    EQU   08H
0009                    DMA     EQU   09H
0020                    OCW2    EQU   20H
0008                    LOADN   EQU   08H
000A                    LSB     EQU   0AH
```

```
0028            MSB     EQU   28H
0227            GETHX   EQU   0227H
0014            MON     EQU   0014H
025C            GETNM   EQU   025CH
0220            GETCH   EQU   0220H
01E8            CO      EQU   01E8H
010F            CNVBN   EQU   010FH
01C2            BREAK   EQU   01C2H
02A1            MILO    EQU   02A1H
02C8            NMOUT   EQU   02C8H
0355            STHLF   EQU   0355H
000D            CR      EQU   0DH
000A            LF      EQU   0AH
0007            BELL    EQU   07H
007F            RUB     EQU   7FH
3FF9            TEMP    EQU   3FF9H
0900                    ORG   0900H
                ;
                ;
                ;  ***********************************************
                ;
                ;   INITIALIZATION OF THE 8259 INTERRUPT PROCESSOR
                ;
                ;  ***********************************************
                ;
0900            INT:
0900    F3              DI              ;DISABLE THE 8080 INTERRUPTS
0901    3EFF            MVI A,ICW1      ;SETS NESTED MODE INTERRUPTS, 4 BYTES APART
0903    D3D8            OUT 0D8H        ;WRITE ICW1 TO 8259
0905    3E08            MVI A,ICW2      ;PUTS CALL TABLE AT 03E0
0907    D3D9            OUT 0D9H        ;WRITE ICW2 TO 8259
0909    FB              EI              ;ENABLE 8080 INTERRUPTS
                ;
                ;
                ;  ***********************************************
                ;
                ;   PROGRAM SELECTION ROUTINE
                ;
                ;  ***********************************************
                ;
090A            SEL:
090A    06C8            MVI B,MSILG ;PRINT MENU
090C    218809          LXI H,MESG1 ;
090F    CD6109          CALL OUTMS  ;
                ;
0912    CD2702  ENTER:  CALL GETHX  ;GET PROGRAM NUMBER
0915    79              MOV A,C     ;MOVE TO ACCUMULATOR
0916    327309          STA LOC     ;STORE FOR LATER USE
0919    FE01            CPI 01H     ;IS IT PROGRAM 1?
091B    CA3D09          JZ INIT     ;IF SO, JUMP TO CZT INITIALIZATION ROUTINE
091E    FE02            CPI 02H     ;IS IT PROGRAM 2?
```

```
0920  CA3D08          JZ INIT         ;IF SO, JUMP TO CZT INITIALIZATION ROUTINE
0923  FE03            CPI 03H         ;IS IT PROGRAM 3?
0925  CA5408          JZ ATOD         ;IF YES, JUMP TO ADC ROUTINE
0928  FE04            CPI 04H         ;IS IT PROGRAM 4?
092A  CACA09          JZ CL           ;IF SO JUMP TO CASSETTE LOADER ROUTINE
092D  FE05            CPI 05H         ;IS IT PROGRAM 5?
092F  CA1709          JZ CSPUN        ;IF SO JUMP TO CASSETTE PUNCH ROUTINE
0932  061C            MVI B,MS2LG     ;IF NONE OF THE ABOVE, PRINT ERROR MESSAGE
0934  21530A          LXI H,MESG2     ;
0937  CD6109          CALL OUTMS      ;
093A  C31208          JMP ENTER       ;RE-ENTER PROGRAM SELECTION
                      ;
                      ;*************************************************
                      ;
                      ; CZT INITIALIZATION ROUTINE
                      ;
                      ;*************************************************
                      ;
093D                  INIT:
093D  CD5809          CALL LDDMA      ;LOAD DMA DESTINATION ADRESS
0940  0623            MVI B,MS3LG     ;PROMPT FOR NUMBER OF RECORDS
0942  216F0A          LXI H,MESG3     ;
0945  CD6109          CALL OUTMS      ;
                      ;
0948  CD2702          CALL GETHX      ;GET NUMBER OF RECORDS
0948  79              MOV A,C         ;MOVE TO ACCUMULATOR
094C  D308            OUT LOADN       ;STORE ON CZT CONTROLLER BOARD
094E  061B            MVI B,MS4LG     ;INDICATE SAMPLE ACCUMULATION
0950  21920A          LXI H,MESG4     ;IS IN PROGRESS
0953  CD6109          CALL OUTMS      ;
0956  76              HLT             ;WAIT FOR INTERRUPT
0957  00              NOP             ;
0958  C33D09          JMP INIT        ;UPON RETURN FROM INTERRUPT, GO BACK TO INIT
                      ;
                      ;*************************************************
                      ;
                      ; LOAD DESTINATION ADRESS ROUTINE
                      ;
                      ;*************************************************
                      ;
0958                  LDDMA:
0958  3E00            MVI A,00        ;SETS LSB OF ADRESS TO 00
095D  D30A            OUT LSB         ;AND MSB TO 60, THEREFORE
095F  3E60            MVI A,60H       ;THE DMA DESTINATION ADRESS
0961  D308            OUT MSB         ;IS 6000H.
0963  C9              RET             ;
                      ;
                      ;*************************************************
```

```
                        ;
                        ;  ADC ROUTINE
                        ;
                        ; ************************************************
                        ;
0864                    ATOD:
0864    CD5B09                  CALL LDDMA      ;LOAD DMA DESTINATION ADRESS
0867    3EFF                    MVI A,0FFH      ;SETS NO. OF RECORDS TO 256
0869    D309                    OUT LOADN       ;LOADS NUMBER OF RECORDS
086B    CDC201                  CALL BREAK      ;SEE IF OPERATOR WANTS OUT
086E    DA1400                  JC MON          ;IF SO GO TO MONITOR
0871    76                      HLT             ;
                        ;
                        ; ************************************************
                        ;
                        ;  SUCCESSFUL INTERRUPT ROUTINE
                        ;
                        ; ************************************************
                        ;
0872                    GOOD:
0872    D309                    OUT DMA         ;START DMA TRANSFER
0874    3A7308                  LDA LOC         ;RECALL WHICH PROGRAM WAS LOADED
0877    FE03                    CPI 03H         ;IS IT 3?
0879    C28408                  JNZ SAMP        ;IF NOT GO TO SAMP
087C    3E20                    MVI A,OCW2      ;IF SO, SEND 8259 A NON-
087E    D3D8                    OUT 0D8H        ;SPECIFIC END OF INTERRUPT (OCW2)
0880    FB                      EI              ;REENABLE 8080 INTERRUPTS
0881    C36409                  JMP ATOD        ;GO BACK TO ADC ROUTINE
0884    FE01            SAMP:   CPI 01H         ;WAS IT PROGRAM 1?
0886    C29109                  JNZ TRANS       ;IF NOT, GO TO TRANS
0889    3E20            BACK:   MVI A,OCW2      ;IF SO, SEND 8259 A NON-
088B    D3D8                    OUT 0D8H        ;SPECIFIC END OF INTERRUPT (OCW2)
088D    FB                      EI              ;REENABLE 8080 INTERRUPTS
088E    C33D09                  JMP INIT        ;GO BACK TO INIT
0891    CDA209          TRANS:  CALL TX         ;PROGRAM SELECTED WAS 2, DO A
0894    C38909                  JMP BACK        ;TRANSFER TO TI, THEN GO TO BACK
                        ;
                        ; ************************************************
                        ;
                        ;  FALSE INTERRUPT ROUTINE
                        ;
                        ; ************************************************
                        ;
0897                    BAD:
0897    0612                    MVI B,MSSLG     ;WRITE FALSE INTERRUPT
0899    21AD0A                  LXI H,MESGS     ;
089C    CD6109                  CALL OUTMS      ;
089F    C31400                  JMP MON         ;GO BACK TO MONITOR
                        ;
```

```
                        ;
                        ;**********************************************
                        ;
                        ;  TRANSFER DATA TO TI ROUTINE
                        ;
                        ;**********************************************
                        ;
03A2                    TX:
03A2    3ECE                    MVI A,0CEH      ;SET UP SERIAL PORT
03A4    D3ED                    OUT 0EDH        ;FOR 9600 BAUD
03A6    3E27                    MVI A,27H       ;
03A8    D3ED                    OUT 0EDH        ;
03AA    217017                  LXI H,5070      ;LOAD START ADDRESS OF DATA
03AD    DBED            PLACE:  IN 0EDH         ;WAIT UNTIL TRANSMITTER READY
03AF    E601                    ANI 01          ;
03B1    CAAD03                  JZ PLACE        ;
03B4    77                      MOV M,A         ;SEND DATA BYTE TO TI
03B5    D3EC                    OUT 0ECH        ;
03B7    23                      INX H           ;GO TO NEXT BYTE
03B8    3E68                    MVI A,68H       ;REPEAT IF NOT LAST ONE
03BA    BC                      CMP H           ;
03BB    C2AD03                  JNZ PLACE       ;
03BE    DBED            WAIT:   IN 0EDH         ;WAIT TILL TRANSMIT BUFFER
03C0    E604                    ANI 04          ;IS EMPTY
03C2    CABE03                  JZ WAIT         ;
03C5    3E40                    MVI A,40H       ;RESET SERIAL PORT
03C7    D3ED                    OUT 0EDH        ;
03C9    C9                      RET             ;RETURN
                        ;
                        ;**********************************************
                        ;
                        ;  CASSETTE LOADER ROUTINE
                        ;
                        ;**********************************************
                        ;
03CA                    CL:
03CA    0636                    MVI B,MSGLG     ;PRINT INSTRUCTIONS FOR PROGRAM
03CC    21BFDA                  LXI H,MESG6     ;
03CF    CD5109                  CALL OUTMS      ;
03D2    3EFF                    MVI A,0FFH      ;SET FOR UPPER NIBBLE
03D4    32F93F                  STA TEMP        ;
03D7    CD2702                  CALL GETHX      ;GET LOAD ADRESS
03DA    50                      MOV D,B         ;PUT IN DE
03DB    59                      MOV E,C         ;
03DC    0E10                    MVI C,10H       ;CS1 TO PLAY
03DE    CDE801                  CALL CO         ;
03E1    0E36                    MVI C,36H       ;
03E3    CDE801                  CALL CO         ;
03E6    0E11                    MVI C,11H       ;
03E8    CDE801                  CALL CO         ;PLAYBACK ON/KEYBOARD OFF
```

```
08EB  CD2002    START: CALL GETCH   ;WAIT FOR START SYMBOL
08EE  FE3E             CPI '>'      ;
08F0  C2EB08           JNZ START    ;
08F3  CD2002    GET:   CALL GETCH   ;GET A CHARACTER
08F6  FE7F             CPI 7FH      ;IGNORE RUB OUTS
08F8  CAF308           JZ GET       ;
08FB  FE3C             CPI '<'      ;STOP IF END SYMBOL
08FD  CA1400           JZ MON       ;GO BACK TO MONITOR
0900  CDDF01           CALL CNVBN   ;CONVERT TO BINARY
0903  4F               MOV C,A      ;
0904  CD5503           CALL STHLF   ;STORE IN MEMORY
0907  3AF93F           LDA TEMP     ;INCREMENT TO NEXT NIBBLE
090A  B7               ORA A        ;
090B  C20F09           JNZ SKIP     ;
090E  13               INX D        ;
090F  EEFF      SKIP:  XRI 0FFH     ;
0911  32F93F           STA TEMP     ;
0914  C3F308           JMP GET      ;REPEAT
                       ;
                       ;*********************************************
                       ;
                       ; CASSETTE PUNCH (RECORD) ROUTINE
                       ;
                       ;*********************************************
                       ;
0917                   CSPUN:
0917  067E             MVI B,MSTLG  ;PRINT OPERATOR INSTRUCTIONS
0919  21F50A           LXI H,MESG   ;
091C  CD6109           CALL OUTMS   ;
091F  0E02             MVI C,02     ;GET START AND STOP ADRESSES
0921  CD5C02           CALL GETNM   ;
0924  0E10             MVI C, 10H   ;CSI TO RECORD
0926  CDE801           CALL CO      ;
0929  0E35             MVI C,35H    ;
092B  CDE801           CALL CO      ;
092E  0E12             MVI C,12H    ;RECORD ON/PRINTER OFF
0930  CDE801           CALL CO      ;
0933  0E3F             MVI C,'>'    ;WRITE START SYMBOL
0935  CDE801           CALL CO      ;
0938  D1               POP D        ;PUT STOP ADRESS IN D E
0939  E1               POP H        ;PUT START ADRESS IN HL
093A  7E        AGAIN: MOV A,M      ;GET A DATA BYTE
093B  CDC802           CALL NMOUT   ;WRITE IT OUT
093E  CDA102           CALL HILO    ;REPEAT TILL DONE
0941  23               INX H        ;
0942  C23A09           JNZ AGAIN    ;
0945  0E3C             MVI C,'<'    ;WRITE END SYMBOL
0947  CDE801           CALL CO      ;
094A  0E13             MVI C,13H    ;WRITE DC3 (PLAYBACK OFF)
094C  CDE801           CALL CO      ;
```

```
8080 MACRO ASSEMBLER, VER 2.4          ERRORS = 0 PAGE 8

094F   0E14          MVI C, 14H   ;RECORD OFF/PRINTER ON
0951   CDE801        CALL CO      ;
0954   0E10          MVI C,10H    ;CSI TO PLAY
0956   CDE801        CALL CO      ;
0959   0E36          MVI C,36H    ;
095B   CDE801        CALL CO      ;
095E   C31403        JMP MON      ;GO TO MONITOR
                     ;
                     ;*************************************************
                     ;
                     ; MESSAGE PRINTING ROUTINE
                     ;
                     ;*************************************************
                     ;
0961                 OUTMS:
0961   4E                MOV C,M      ;MOVE CHARACTER TO C
0962   79                MOV A,C      ;AND TO A
0963   FE0D              CPI CR       ;SEE IF CHARACTER IS A RETURN
0965   CA7109            JZ CRRUB     ;IF SO, JUMP TO CRRUB
0968   CDE801            CALL CO      ;IF NOT OUTPUT CHARACTER
096B   23                INX H        ;INCREMENT POINTER
096C   05                DCR B        ;DECREMENT COUNTER
096D   C26109            JNZ OUTMS    ;JUMP BACK IF NOT FINISHED
0970   C9                RET          ;IF FINISHED , RETURN
0971   CDE801        CRRUB: CALL CO   ;OUTPUT CARRIAGE RETURN
0974   0E7F              MVI C,RUB    ;PUT ASCII RUB OUT IN C
0976   CDE801            CALL CO      ;OUTPUT 5 RUBOUTS,
0979   CDE801            CALL CO      ;IT PREVENTS THE SILENT 700
097C   CDE801            CALL CO      ;FROM MISSING ANY CHARACTERS
097F   CDE801            CALL CO      ;DUE TO ITS SLOW
0982   CDE801            CALL CO      ;CARRIAGE RETURN
0985   23                INX H        ;INCREMENT POINTER
0986   05                DCR B        ;DECREMENT  COUNTER
0987   C26109            JNZ OUTMS    ;JUMB BACK IF NOT FINISHED
098A   C9                RET          ;IF FINISHED RETURN
                     ;
                     ;********************* MESSAGES *********************
                     ;
                     ;
098B                 MESG1:
098B   0D0A0A53           DB CR,LF,LF,'SELECT PROGRAM:',CR,LF
098F   454C4543
0993   54205052
0997   4F475241
099B   4D3A0D0A
099F   31203D20           DB '1 = SAMPLE ACCUMULATION AND DMA TRANSFER',CR,LF
09A3   53414D50
09A7   4C452041
09AB   4343554D
```

```
09AF  554C4154
09B3  494F4320
09B7  414E4420
09BB  444D4120
09BF  5452414E
09C3  53464552
09C7  3D0A
09C9  32203D20          DB '2 = SAMPLE ACCUMULATION,DMA TRANSFER, AND TRANSFER '3
09CD  53414D50
09D1  4C452041
09D5  43435540
09D9  554C415A
09DD  494F4E2C
09E1  444D4120
09E5  5452414E
09E9  53464552
09ED  2C20414F
09F1  44205452
09F5  414F5346
09F9  45522054
09FD  4F205449
0A01  0D0A3320          DB CR,LF,'3 = ADC ALIGN',CR,LF
0A05  3D204144
0A09  4320414C
0A0D  49474E0D
0A11  0A
0A12  34203D20          DB '4 = CASSETTE LOAD',CR,LF
0A16  43415353
0A1A  45544445
0A1E  204C4F41
0A22  440D0A
0A25  35203D20          DB '5 = CASSETTE PUNCH',CR,LF
0A29  43415353
0A2D  45544445
0A31  2050554E
0A35  43480D0A
0A39  45534320          DB 'ESC = RETURN TO MONITOR',CR,LF,BELL
0A3D  3D205245
0A41  5455524E
0A45  20544F20
0A49  4D4F4E49
0A4D  544F520D
0A51  0A07
00C8            MS1LG   EQU $-MESG1
                ;
                ;
0A53            MESG2:
0A53  0D494E56          DB CR,'INVALID ENTRY,TRY AGAIN.',CR,LF,BELL
0A57  414C4944
0A5B  20454E54
0A5F  52592C54
```

```
0A63   52592041
0A67   4741494E
0A6B   2E0D0A07
0031C              MS2LG   EQU $-MESG2
                   ;
                   ;
0A6F               MESG3:
0A6F   0D4C4F41            DB CR,'LOAD NUMBER OF RECORDS  (2 HEX)  ',BELL
0A73   44204E55
0A77   4D424552
0A7B   204F4620
0A7F   5245434F
0A83   52445320
0A87   20203220
0A8B   48455929
0A8F   202007
0023               MS3LG   EQU $-MESG3
                   ;
                   ;
0A92               MESG4:
0A92   0D414343            DB CR,'ACCUMULATION IN PROGRESS',CR,LF
0A96   554D554C
0A9A   4154494F
0A9E   4E20494E
0AA2   2050524F
0AA6   47524553
0AAA   530D0A
0018               MS4LG   EQU $-MESG4
                   ;
                   ;
0AAD               MESG5:
0AAD   0D46414C            DB CR,'FALSE INTERRUPT',CR,LF
0AB1   53452049
0AB5   4E544552
0AB9   52555054
0ABD   0D0A
0012               MS5LG EQU $-MESG5
                   ;
                   ;
0ABF               MESG6:
0ABF   0D454E54            DB CR,'ENTER ADDRESS WHERE PROGRAM IS TO BE LOADED (4 HEX)
0AC3   45522041
0AC7   44445245
0ACB   53532057
0ACF   48455245
0AD3   2050524F
0AD7   4752414D
0ADB   20495320
0ADF   544F2042
0AE3   45204C4F
0AE7   41444544
```

```
0AE0    20283420
0AE7    45455829
0AF3    2007
0036                MSGLG   EQU$-MESG6
                    ;
                    ;
0AF5                MESG7:
0AF5    0D454E54            DB CR,'ENTER STARTING ADDRESS (4 HEX) THEN SPACE AND ENTER'
0AF9    45522053
0AFD    54415254
0B01    494E4720
0B05    41444452
0B09    45535320
0B0D    28342048
0B11    45582920
0B15    5448454E
0B19    20535041
0B1D    43452041
0B21    4E442045
0B25    4E544552
0B29    0D0A
0B2B    454E4449            DB 'ENDING ADDRESS OF DATA TO BE RECORDED.',CR,LF
0B2F    4E472041
0B33    44445245
0B37    5353204F
0B3B    46204441
0B3F    54412054
0B43    4F204245
0B47    20524543
0B4B    4F524445
0B4F    442E0D0A
0B53    5455524E            DB 'TURN ON RECORD AND HIT RETURN',CR,LF,BELL
0B57    204F4E20
0B5B    52454 34F
0B5F    52442041
0B63    4E442048
0B67    49542052
0B6B    45545552
0B6F    4E0D0407
0B7E                MS7LG   EQU $-MESG7
                    ;
                    ;
0B7E                LOC:    DS 01
                    ;
                    ; ***********************************************
                    ;
                    ;  INTERRUPT JUMP TABLE
                    ;
                    ; ***********************************************
                    ;
```

```
08E0                    ORG   0BE0H
08E0              TABLE:
08E0  C39709            JMP BAD
08E3  00                NOP
08E4  C37209            JMP GOOD
08E7  00                NOP
08E8  C39709            JMP BAD
08EB  00                NOP
08EC  C39709            JMP BAD
08EF  00                NOP
08F0  C39709            JMP BAD
08F3  00                NOP
08F4  C39709            JMP BAD
08F7  00                NOP
08F8  C39709            JMP BAD
08FB  00                NOP
08FC  C39709            JMP BAD
08FF  00                NOP
                        END
NO PROGRAM ERRORS
```

SYMBOL TABLE

* 01

A	0007	AGAIN	093A	ATCD	0864	B	0000
BACK	0889	BAD	0897	BELL	0007	BREAK	01C2
C	0001	CL	09CA	CNVBN	01DF	CO	01E8
CR	000D	CRRUB	0971	CSFUN	0917	D	0002
DMA	0009	E	0003	ENTER	0812	GET	08F3
GETCH	0220	GETHX	0227	GETNM	025C	GOOD	0872
H	0004	HILO	02A1	ICW1	00FF	ICW2	0008
INIT	083D	INT	0900 *	L	0005	LDDMA	085B
LF	000A	LOADN	0008	LOC	08F3	LSB	000A
M	0005	MESG1	0988	MESG2	0A53	MESG3	0A6F
MESG4	0A92	MESG5	0AAD	MESG6	0ABF	MESG7	0AF5
MSN	0014	MS1LG	00C8	MS2LG	001C	MS3LG	0023
MS4LG	0018	MS5LG	0012	MS6LG	0036	MS7LG	007E
MSB	000B	NMOUT	09C8	OCW2	0020	OUTMS	0961
PLACE	08AD	PSW	0006	PUB	027F	SAMP	0884
SEL	08DA	SKIP	090F	SP	0004	START	08EB
STHLF	0355	TABLE	0850 *	TEMP	3FF9	TRANS	0891
TX	09A2	WAIT	098E				

APPENDIX H

CONTROLLER ROM CONTENTS

	HEX Equivalent		R/$\bar{W}$	OD	WLAT	RLAT	START CONVERT	$\bar{S}$/H	LD/R	LDDR	CZT CLK	T1CL	T2CL	T3CL	T4CL	T1K	T3R	$\overline{\text{DIEN}}$	HEX Equivalent		
0	8	4	1	0	0	0	0	1	0	0	0	1	1	0	1	0	1	1	6	B	0
1	8	4	1	0	0	0	0	1	0	0	0	1	1	0	1	0	1	1	6	B	1
2	8	4	1	0	0	0	0	1	0	0	0	1	1	0	1	0	1	1	6	B	2
3	8	4	1	0	0	0	0	1	0	0	0	1	1	0	1	0	1	1	6	B	3
4	8	4	1	0	0	0	0	1	0	0	0	1	1	0	1	0	1	1	6	B	4
5	8	4	1	0	0	0	0	1	0	0	0	1	1	0	1	0	1	1	6	B	5
6	8	4	1	0	0	0	0	1	0	0	0	1	1	0	1	0	1	1	6	B	6
7	8	4	1	0	0	0	0	1	0	0	0	1	1	0	1	0	1	1	6	B	7
8	8	4	1	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	2	7	8
9	8	4	1	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	2	7	9
10	8	4	1	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	2	7	10
11	8	4	1	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	2	7	11
12	8	4	1	0	0	0	0	1	0	0	0	0	1	1	0	1	1	1	3	7	12
13	8	4	1	0	0	0	0	1	0	0	0	0	1	1	0	1	1	1	3	7	13
14	8	6	1	0	0	0	0	1	1	0	0	0	1	1	0	1	1	1	3	7	14
15	8	4	1	0	0	0	0	1	0	0	0	0	1	1	0	1	1	1	3	7	15
16	8	5	1	0	0	0	0	1	0	1	0	0	1	1	1	1	1	1	3	F	16
17	8	4	1	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	3	F	17
18	9	4	1	0	0	1	0	1	0	0	0	0	1	1	1	1	1	1	3	F	18
19	9	4	1	0	0	1	0	1	0	0	0	0	1	1	1	1	1	1	3	F	19
20	8	4	1	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	3	F	20
21	8	4	1	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	3	F	21

	HEX Equivalent		R/$\bar{W}$	OD	WLAT	RLAT	START CONVERT	S/$\bar{H}$	LD/R	LDDR	CZT CLK	T1CL	T2CL	T3CL	T4CL	T1K	T3R	$\overline{DIEN}$	HEX Equivalent		
22	8	4	1	0	0	0	0	1	0	0	0	0	1	1	1	1	1	0	3	E	22
23	8	4	1	0	0	0	0	1	0	0	0	0	1	1	1	1	1	0	3	E	23
24	A	4	1	0	1	0	0	1	0	0	0	0	1	1	1	1	1	0	3	E	24
25	A	4	1	0	1	0	0	1	0	0	0	0	1	1	1	1	1	0	3	E	25
26	8	4	1	0	0	0	0	1	0	0	0	0	1	1	1	1	1	0	3	E	26
27	8	4	1	0	0	0	0	1	0	0	0	0	1	1	1	1	1	0	3	E	27
28	C	4	1	1	0	0	0	1	0	0	0	0	1	1	1	1	0	0	3	C	28
29	C	4	1	1	0	0	0	1	0	0	0	0	1	1	1	1	0	0	3	C	29
30	4	4	0	1	0	0	0	1	0	0	0	0	1	1	1	1	0	0	3	C	30
31	4	4	0	1	0	0	0	1	0	0	0	0	1	1	1	1	0	0	3	C	31
32	C	4	1	1	0	0	0	1	0	0	0	0	1	1	1	1	0	0	3	C	32
33	C	4	1	1	0	0	0	1	0	0	0	0	1	1	1	1	0	0	3	C	33
34	8	4	1	0	0	0	0	1	0	0	0	0	1	1	1	1	0	0	3	C	34
35	8	4	1	0	0	0	0	1	0	0	0	0	1	1	1	1	0	0	3	C	35
36	8	4	1	0	0	0	0	1	0	0	1	0	1	1	1	1	1	1	B	F	36
37	8	4	1	0	0	0	0	1	0	0	1	0	1	1	1	1	1	1	B	F	37
38	8	4	1	0	0	0	0	1	0	0	1	0	1	1	1	1	1	1	B	F	38
39	8	4	1	0	0	0	0	1	0	0	1	0	1	1	1	1	1	1	B	F	39
40	8	4	1	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	0	F	40
41	8	4	1	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	0	F	41
42	8	4	1	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	0	F	42

	HEX Equivalent		R/$\bar{W}$	OD	WLAT	RLAT	START CONVERT	$\bar{S}$/H	LD/R	LDDR	CZT CLK	T1CK	T2CL	T3CL	T4CL	T1K	T3R	$\overline{DTEN}$	HEX Equivalent		
43	8	4	1	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	0	F	43
44	8	4	1	0	0	0	0	1	0	0	0	1	0	0	1	1	1	1	4	F	44
45	8	4	1	0	0	0	0	1	0	0	0	1	0	0	1	1	1	1	4	F	45
46	8	4	1	0	0	0	0	1	0	0	0	1	0	0	1	1	1	1	4	F	46
47	8	4	1	0	0	0	0	1	0	0	0	1	0	0	1	1	1	1	4	F	47
48	8	4	1	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	6	F	48
49	8	4	1	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	6	F	49
50	8	4	1	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	6	F	50
51	8	0	1	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	6	F	51
52	8	0	1	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	6	F	52
53	8	0	1	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	6	F	53
54	8	0	1	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	6	F	54
55	8	0	1	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	6	F	55
56	8	0	1	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	6	F	56
57	8	0	1	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	6	F	57
58	8	0	1	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	6	F	58
59	8	C	1	0	0	0	1	1	0	0	0	1	1	0	1	1	1	1	6	F	59
60	8	4	1	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	6	F	60
61	8	4	1	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	6	F	61
62	8	4	1	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	6	F	62
63	8	4	1	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	6	F	63

The REMOTE SENSING CENTER was established by authority of the Board of Directors of the Texas A&M University System on February 27, 1968. The CENTER is a consortium of four colleges of the University; Agriculture, Engineering, Geosciences, and Science. This unique organization concentrates on the development and utilization of remote sensing techniques and technology for a broad range of applications to the betterment of mankind.